The Dissertation Committee for Socratis Petrides
certifies that this is the approved version of the following dissertation:

Adaptive multilevel solvers for the discontinuous Petrov–Galerkin method with an emphasis on high-frequency wave propagation problems

Committee:

Leszek F. Demkowicz, Supervisor

George Biros

Tan Bui-Thanh

Björn Engquist

Jay Gopalakrishnan

Chris Simmons

Adaptive multilevel solvers for the discontinuous Petrov–Galerkin method with an emphasis on high-frequency wave propagation problems

by

Socratis Petrides

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2019

To my beautiful wife

Acknowledgments

This work would have not been possible without the help and support of several people from both my academic and personal life. First and foremost, I would like to express my deep gratitude to my research advisor Dr. Leszek Demkowicz for his patient guidance and constant encouragement at every step of this journey. Working under his supervision, enabled me to learn a great deal about both theory and practice of finite element methods, solvers and numerical analysis in general; skills that opened a lot of new doors in my life. I would also like to thank him for the unforgettable coding sessions and enjoyable discussions (academic and otherwise) in his office along with good coffee, for all the opportunities he has given me to present my work worldwide, for the Christmas parties that made both me and my wife feel at home and for his invaluable help during my job applications. He is the best teacher and mentor anyone could hope for.

I wish to thank my committee members: Dr. George Biros, Dr. Tan Bui-Thanh, Dr. Björn Engquist, Dr. Jay Gopalakrishnan and Dr. Chris Simmons for agreeing to serve on my dissertation committee and for their valuable feedback regarding this work. I am especially grateful to Dr. Jay Gopalakrishnan for his suggestions and insightful comments on the theory of preconditioning. His help has been crucial to the completion of this work.

This work has been supported by the Air Force Office of Scientific Research (AFOSR). I would like to extend my thanks and appreciation to the officers of the AFSOR Computational Mathematics program, Dr. Fariba Fahroo and Dr. Jean-Luc Cambier for their continuing support.

I am indebted to many people from the Institute for Computational Engineering and Sciences (ICES). First, I would like to thank my close associates, the DPG "dream team", Dr. Federico Fuentes, Stefan Henneking, Dr. Brendan Keith, Jaime Mora, Dr. Sriram Nagaraj and Dr. Ali Vaziri for our wonderful collaboration over the past few years. I consider myself very

fortunate to have worked with these great mathematicians, engineers and computer scientists. In addition, I thank my fellow classmates, especially Sam Estes, for our technical discussions. They really enhanced my understanding on fundamental mathematical concepts and that was key to my research development. Lastly, I would like to express my great appreciation to Stephanie Rodriguez and Lauren Constant who would always go out of their way to assist me whenever I needed help.

Next, I wish to thank my friends who have been by my side throughout these years. My special thanks go to Alex and Stelana for proofreading this document. I also thank my former boss and a very good friend Marios Hadjivassiliou, who gave me the nudge to return back to Austin and pursue my dreams. Finally, I am very grateful to my extended family back home, particularly my parents-in-law, my sister and my parents, who have been very encouraging and supportive in many different ways. Mom, dad, I owe so much to you both. Thank you for making me who I am today.

This work is dedicated to the most important person in my life, my loving wife Kyriaki. Without her true love and devotion all these years none of this would have been possible. She has been the driving force that kept me moving forward. Thank you for always helping me think clearly, for pulling me up when I was down, for being so understanding during my long hours of studying and for always believing in me. Thank you for not letting me settle for less. Undoubtedly, I couldn't have made it without you. I love you!

Thank you all for this amazing journey.

Adaptive multilevel solvers for the discontinuous Petrov–Galerkin method with an emphasis on high-frequency wave propagation problems

by

Socratis Petrides
The University of Texas at Austin, 2019

Supervisor: Leszek F. Demkowicz

This dissertation focuses on the development of fast and efficient solution schemes for the simulation of challenging problems in wave propagation phenomena. In particular, emphasis is given in high frequency acoustic and electromagnetic problems which are characterized by localized solutions. This kind of simulations are essential in various applications, such as ultrasonic testing, laser scanning and modeling of optical laser amplifiers.

In wave simulations, the computational cost of any numerical method, is directly related to the frequency. In the high-frequency regime very fine meshes have to be used in order to satisfy the Nyquist criterion and overcome the pollution effect. This often leads to prohibitively expensive problems. Numerical methods based on standard Galerkin discretizations, lack of pre-asymptotic discrete stability and therefore adaptive mesh refinement strategies are usually inefficient. Additionally, the indefinite nature of the wave operator makes state of the art preconditioning techniques, such as multigrid, unreliable.

In this work, a promising alternative approach is followed within the framework of the discontinuous Petrov–Galerkin (DPG) method. The DPG method offers numerous advantages for our problems of interest. First and foremost, it offers mesh and frequency independent discrete stability even in the pre-asymptotic region. This is made possible by computing on the fly an optimal test space as a function of the trial space. Secondly, it provides a built-in local

error indicator that can be used to drive adaptive refinements. Combining these two properties together, reliable adaptive refinement strategies are possible which can be initiated from very coarse meshes. Lastly, the DPG method can be viewed as a minimum residual method, and therefore it always delivers symmetric (Hermitian) positive definite stiffness matrix. This is a desirable advantage when it comes to the design of iterative solution algorithms. Conjugate Gradient based solvers can be employed which can be accelerated by domain decomposition (one- or multi- level) preconditioners for symmetric positive definite systems.

Driven by the aforementioned properties of the DPG method, an adaptive multigrid preconditioning technology is developed that is applicable for a wide range of boundary value problems. Unlike standard multigrid techniques, our preconditioner involves trace spaces defined on the mesh skeleton, and it is suitable for adaptive *hp*-meshes. Integration of the iterative solver within the DPG adaptive procedure turns out to be crucial in the simulation of high frequency wave problems. A collection of numerical experiments for the solution of linear acoustics and Maxwell equations demonstrate the efficiency of this technology, where under certain circumstances uniform convergence with respect to the mesh size, the polynomial order and the frequency can be achieved. The construction is complemented with theoretical estimates for the condition number in the one-level setting.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Accurate and efficient numerical simulation of wave propagation phenomena is still an active research area in the computational science community. Practical applications can be found in numerous fields including industry, medicine, communications and military. This work is motivated by applications in the high-frequency regime, such as *ultrasonic testing*, *laser scanning* (also known as LIDAR) and *optical laser amplifiers*.

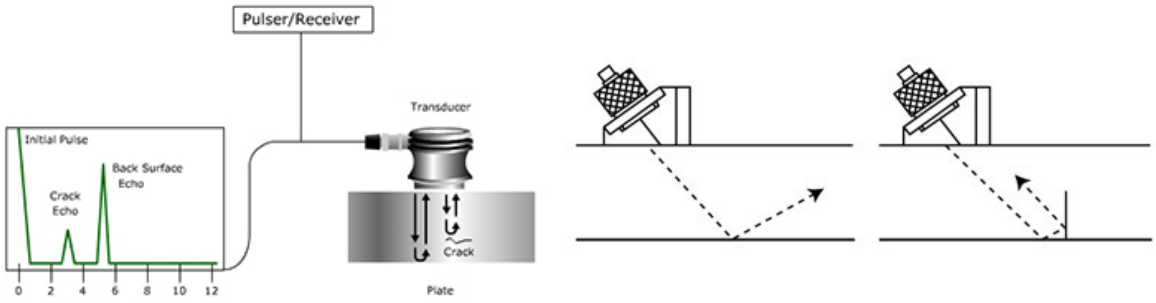


Figure 1.1: Ultrasonic testing: a transducer generates an high frequency sound wave, which propagates through the material, and then by monitoring the intensity of reflection of the wave, the material can be examined for possible flaws (left figure retrieved January 10, 2019, from http://www.sdindt.com/Ultrasonic-Testing.html, right figure retrieved January 10, 2019, from https://www.olympus-ims.com/zh/ resources/white-papers/an-introduction-to-angle-beam-assemblies/)

Ultrasonic testing techniques are popular in industry when examining the properties of a material or detecting possible flaws in it [69]. In practice, a transducer generates an ultrasonic wave, which propagates through the examined material, and then by monitoring the intensity of reflection and/or the attenuation of the wave, the material can be characterized

(see Figure 1.1). Laser scanning is a remote sensing method, that measures distance to a target. Unlike ultrasonic testing techniques, in laser scanning the target is illuminated by a laser light, and the reflected waves are measured by a sensor, in order to construct a digital representation of the target. LIDARs are commonly used for the construction of high-resolution maps and for control and navigation of autonomous cars. Lastly, as the name suggests, an optical laser amplifier is a device used to amplify an optical signal. They are commonly used in military and industrial applications. Our research group, in collaboration with the Air Force Research Lab, is working on the simulation of high-power fiber lasers based on the *Raman gain model*, work that is described in detail in [97, 96].

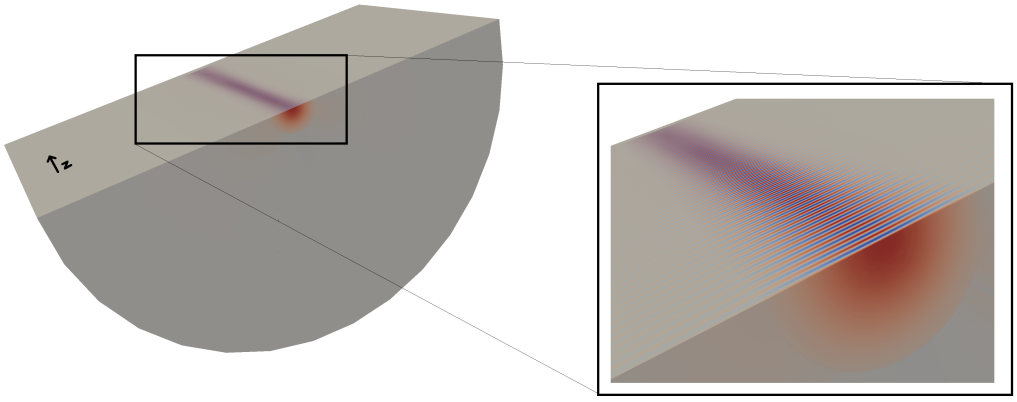


Figure 1.2: Simulation of fiber laser using the Raman gain model (figure retrieved January 10, 2019, from [97]).

1.2 Objective

Driven by the aforementioned applications, this dissertation focuses on the development of fast and efficient solvers for the simulation of acoustic and electromagnetic wave propagation phenomena. We are particularly interested in the time-harmonic form of linear acoustics and Maxwell equations. Emphasis is given to the simulation of high frequency beams and problems with singular solutions. The key point of our work is the construction of robust and effective

multilevel preconditioners for the discontinuous Petrov–Galerkin (DPG) method and how they can be integrated with automatic *hp*-adaptive strategies.

1.3 Background

1.3.1 Time-harmonic wave equations

Time-harmonic (also known as *steady state*) formulations of the time dependent wave equations can be derived under the assumption that the source varies sinusoidally in time with a single frequency. Under this assumption, using phasor analysis, the time-harmonic formulations for acoustics and Maxwell equations are derived below.

1.3.1.1 Linear acoustics

We consider the differential isentropic form of the compressible Euler equations, expressed in terms of density ρ, pressure p and velocity u, i.e,

$$\begin{aligned} \text{Conservation of mass:} \quad & \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho u) = 0 \\ \text{Linear momentum:} \quad & \rho \frac{\partial u}{\partial t} + \rho u \cdot \nabla u + \nabla p = 0 \end{aligned} \quad , \tag{1.1}$$

In order to obtain the classical linear equations, we linearize around the hydrostatic equilibrium position $\rho = \rho_0$, $u = 0$. Consider a perturbation of the solution around the equilibrium position, $\rho = \rho_0 + \delta\rho$ and $u = \delta u$. Assuming that the perturbations are small, we can neglect the higher order terms. Additionally, for the isentropic flow, the pressure is related with the density through an algebraic relation, $p = p(\rho)$, and so linearizing around the equilibrium position gives

$$p = p(\rho_0) + \frac{dp}{d\rho}(\rho_0)\delta\rho + \dots$$

We now define by $p_0 := p(\rho_0)$ the hydrostatic pressure, and identify by $c_0 = \sqrt{\frac{dp}{d\rho}(\rho_0)}$, the speed of sound in the acoustic medium. If we now linearize (1.1) we arrive at:

$$\begin{aligned} \frac{\partial(\delta\rho)}{\partial t} + \rho_0 \operatorname{div} \delta u = 0 \\ \rho_0 \frac{\partial(\delta u)}{\partial t} + \nabla \delta p = 0 \end{aligned}$$

where $\delta p = c_0^2 \delta\rho$ is the perturbation in the pressure. Eliminating density ρ, and with some abuse of notation (dropping the δ symbol) we get

$$\begin{aligned} \frac{1}{c_0^2}\frac{\partial p}{\partial t} + \rho_0 \operatorname{div} u &= 0 \\ \rho_0 \frac{\partial u}{\partial t} + \nabla p &= 0 \end{aligned}$$

Assuming now ansatz in time, we can derive the time-harmonic equations. Let

$$\begin{aligned} p(x,t) &= e^{i\omega t}\hat{p}(x) \\ u(x,t) &= e^{i\omega t}\hat{u}(x). \end{aligned}$$

and let ω be the *angular frequency*. Then, the time-harmonic equations are given by

$$\begin{aligned} \frac{1}{c_0^2} i\omega p + \rho_0 \operatorname{div} u &= 0 \\ \rho_0 i\omega u + \nabla p &= 0 \end{aligned}$$

where for simplicity we dropped the hats ($\hat{}$). After non-dimensionalization, we arrive at:

$$\begin{aligned} i\omega p + \operatorname{div} u &= 0 \\ i\omega u + \nabla p &= 0 \end{aligned} \tag{1.2}$$

Note that by solving for u in the second equation and substituting to the first, (1.2) reduces to the Helmholtz equation:

$$-\Delta p - \omega^2 p = 0$$

A boundary value problem (BVP) can be derived from (1.2) by considering a bounded domain Ω and appropriate boundary conditions. For instance, for general right hand sides and hard and soft boundary conditions the BVP reads:

$$\begin{cases} i\omega p + \operatorname{div} u = f_1, & \text{in } \Omega \\ i\omega u + \nabla p = f_2, & \text{in } \Omega \\ p = 0, & \text{on } \Gamma_1 \\ u \cdot n = 0, & \text{on } \Gamma_2 \end{cases} \tag{1.3}$$

Here, Γ_1, Γ_2 are parts of the boundary $\partial\Omega$ such that $\Gamma_2 = \partial\Omega \setminus \Gamma_1$ and n is the outward unit normal.

1.3.1.2 Electromagnetics - Maxwell equations

Maxwell's equations are the governing equations of all electromagnetic phenomena. They consist of five equations that describe precisely how electric and magnetic fields are generated from their sources: electric charges and currents. These equations in their transient form are:

$$\begin{aligned}
\text{Gauss (electric law)}: &\quad \nabla \cdot \boldsymbol{D} &&= \rho^{\mathsf{imp}} + \rho, \\
\text{Gauss (magnetic law)}: &\quad \nabla \cdot \boldsymbol{B} &&= 0, \\
\text{Faraday}: &\quad \nabla \times \boldsymbol{E} &&= -\frac{\partial \boldsymbol{B}}{\partial t}, \\
\text{Ampere}: &\quad \nabla \times \boldsymbol{H} &&= \frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{J}^{\mathsf{imp}} + \boldsymbol{J}, \\
\text{Continuity equation}: &\quad \frac{\partial \rho}{\partial t} &&= -\nabla \cdot \boldsymbol{J}
\end{aligned}$$

where $\boldsymbol{D}$ is the electric flux density, ρ^{imp} is the impressed charge, ρ is the free charge density, $\boldsymbol{B}$ is the magnetic flux density, $\boldsymbol{E}$ is the electric field, $\boldsymbol{H}$ is the magnetic field, $\boldsymbol{J}^{\mathsf{imp}}$ is the impressed current and $\boldsymbol{J}$ is free electric current density. Assuming linear isotropic materials, the following constitutive relations hold:

$$\boldsymbol{D} = \epsilon \boldsymbol{E}, \quad \boldsymbol{B} = \mu \boldsymbol{H}, \quad \boldsymbol{J} = \sigma \boldsymbol{E}$$

where ϵ, μ and σ are the permitivity, permeability and conductivity of the material, respectively. We can now eliminate the free charge ρ, use the constitutive relations, and apply Fourier transform, to derive the time-harmonic form of Maxwell's equations for the electric and magnetic field:

$$\begin{aligned}
\text{Faraday}: &\quad \nabla \times \hat{\boldsymbol{E}} + i\omega\mu \hat{\boldsymbol{H}} &&= 0, \\
\text{Ampere}: &\quad \nabla \times \hat{\boldsymbol{H}} - (i\omega\epsilon + \sigma)\hat{\boldsymbol{E}} &&= \hat{\boldsymbol{J}}^{\mathsf{imp}},
\end{aligned} \tag{1.4}$$

where the hat ($\hat{}$) denotes the Fourier transform (phasors). Note that the compatibility condition $\nabla \cdot \hat{\boldsymbol{J}}^{\mathsf{imp}} = i\omega\rho^{\mathsf{imp}}$ must be satisfied. The time harmonic-form of Gauss electric and magnetic laws can be derived by taking the divergence of both sides of Ampere's and Faraday's law in (1.4), respectively. A BVP can be derived in a similar way to the derivation of (1.3).

1.3.2 Galerkin methods for high frequency wave propagation problems

There is a wide range of numerical methods for the simulation of high-frequency wave propagations problems. For inhomogeneous materials and complex geometries the state of the art methods are based on Finite Element (commonly referred as standard Galerkin) discretizations. Unfortunately, standard Galerkin methods suffer from the pollution effect [6], especially in the lower order case. In simple terms, as the frequency grows, the numerical solution becomes out of phase, even if the Nyquist criterion is met [78, 77] (fixed number of elements per wavelength). The pollution can be controlled, by using finer meshes or a higher order of approximation [93]. However, this often leads to computationally intractable problems, because the linear system is either very large or badly-conditioned.

Other approaches include the partition of unity methods [5, 91], discontinuous Galerkin methods [92, 24], ultraweak variational formulations (UWVF) [76, 21], plane waves methods (PWDG) [75, 114], interior penalty methods with complex stabilization [48, 49], etc. Even though these methods are often significantly more efficient than the standard Galerkin method, in the multidimensional case the pollution effect is unavoidable [6]. In addition, these methods are only asymptotically stable, and therefore for high frequencies, adaptive schemes work only if they start from a significantly fine initial mesh. Alternative approaches are based on minimum residual methods for the first order system (FOSLS) [86, 94, 10], which provide unconditionally stable discretizations. They experience however, very dissipative behavior.

1.3.3 Linear solvers

For Galerkin discretizations, state of the art sparse direct solvers are very efficient for the solution of two dimensional problems. Even for very large frequencies it is difficult to compete with direct solvers that exploit sparsity patterns and minimize the bandwidth of the stiffness matrix, since under certain circumstances, they can achieve nearly linear complexity. However, in three dimensional computations, this is not the case, since the computational complexity grows quadratically with the number of unknowns.

Standard iterative solvers, being the only alternative, do not behave very well when it

comes to the solution of high-frequency wave equations. The main difficulty arises from the indefiniteness and the conditioning of the linear system. For standard Galerkin discretizations one must use GMRES, with a preconditioner which shifts the spectrum to the positive half plane. The resulting solver is not always reliable, especially when the grid is very coarse. Multigrid techniques loose their efficiency, since in order to converge, they usually require expensive coarse grid solves because the coarse mesh has to be "fine enough". In general, there are numerous approaches for designing fast and robust iterative solvers for high frequency wave propagation problems [45, 47, 58]. In particular, well-known attempts include but are not limited to, shifted Laplacian techniques [47, 111, 59], domain decomposition methods [85, 63, 16, 112], multigrid methods [113, 64, 18], the fast sweeping preconditioner [44, 43, 116], the method of polarized traces [121], and stabilized FEM techniques based on artificial absorption [57, 12]. Nevertheless, developing frequency-independent iterative solvers, that also exploit efficiently today's multi-core computer architectures, is still an open research problem in the scientific community.

1.3.4 The discontinuous Petrov–Galerkin (DPG) method

The DPG method has been gaining much attention over recent years. It was originally developed almost ten years ago by Demkowicz and Gopalakrishnan [27, 28, 35] for the so called *ultraweak formulation*. The ultraweak formulation is derived by considering the first order system of equations and passing all the regularity to the test space though integration by parts. It was later realized that other formulations could be considered without using a first order reformulation [29]. In fact, more recently it has been shown that the DPG method can be applied to any well posed variational formulation [19].

As the name suggests, it is a Petrov–Galerkin scheme since it uses a non-symmetric functional setting, i.e, trial and test spaces are not the same. The novelty of the method comes from the fact that the test functions are computed on the fly, in such a way that discrete stability is inherited from the continuous stability of the original problem. In other words, the trial and test space are not predefined; on the contrary, given a trial space the *optimal test*

space is computed though the so called *trial to test operator*. This method is called the *ideal* DPG method. In real computations, the trial to test operator and the optimal test functions are approximated, resulting in the so called *practical DPG method* [65].

The need for computational efficiency led to the use of *broken* (discontinuous) test spaces at the expense of introducing additional interface unknowns (Lagrange multipliers). The additional cost of approximating the test functions is now reduced to the element level, making the method more competitive. Carstensen et al. in [19], show that the resulting variational formulation with broken test spaces remains well posed with a mesh-independent stability constant of the same order as the inf-sup constant for the original problem.

As it will be demonstrated in Chapter 2, the DPG method admits two additional reformulations. First it can be reformulated as a minimum residual method where the residual is minimized in the dual norm. Consequently, the resulting DPG system is always symmetric (Hermitian) and positive definite. This is an important advantage when designing preconditioners for iterative solvers. Notice that a different choice of the norm on the test space results in a completely different method. Secondly, the DPG method can be written as a mixed method, where simultaneously a solution is sought for the original unknown and the *error representation function*. The norm of the error representation function provides an inexpensive built-in error indicator and therefore the possibility of efficient and reliable adaptive schemes.

In summary, the most important advantages of the DPG method are the following: a) it has a mathematically sound functional analysis background, providing additional comfort in computations, b) it is extremely general in the sense that it can be applied to any well posed variational formulation, c) it provides unconditional discrete stability, d) it always delivers symmetric (Hermitian) positive definite stiffness matrices and e) it comes with a built-in error indicator. However, it has been sometimes criticized for its overall cost. It is a fact that the DPG method delivers linear systems of approximately twice the size compared to the standard Galerkin linear system. Note though, that this is independent of the dimension of the problem. Admittedly, the DPG method might not be the most favorable choice when it comes to simple elliptic problems (for instant solving Poisson equation in a square). The superiority of the

DPG method is apparent in more challenging problems usually when adaptivity is crucial (e.g. problems with boundary layers, shocks, singularities etc.)

Over the past few years several research studies have been devoted to the further theoretical development of the DPG method and its application to engineering problems. Among many others, the DPG method has been applied to fluid [109, 22, 40, 107, 82], convection-diffusion [14, 36, 23], singular perturbation [100, 73, 54, 56, 72], elasticity [20, 81, 51, 50] and wave propagation problems [106, 33, 97, 62, 122]. DPG *space-time* discretizations have been successfully developed for Navier-Stokes, acoustics and Schrödinger equations [41, 42, 67, 68, 46, 34]. Other related works involve coupling different formulations of the DPG method [52], coupling DPG with other discretization methods [54, 71, 73, 55], polygonal elements [3], fast integration techniques [95], goal oriented adaptive strategies and the DPG* method [79, 84, 32, 80] and DPG preconditioners [8, 66, 7, 87, 106, 108].

1.4 Achievements of this dissertation

The main accomplishment of this work is twofold. First, we have developed a general multigrid technology for the DPG method that is suitable for a wide range of boundary value problems. It can be applied to any well posed variational formulation as long as it can be discretized using the energy spaces of the exact sequence

$$H^1 \xrightarrow{\nabla} H(\mathrm{curl}) \xrightarrow{\nabla\times} H(\mathrm{div}) \xrightarrow{\nabla\cdot} L^2.$$

Additionally, we have developed a theoretical convergence analysis for the case of the one level Schwarz preconditioner for the linear acoustics problem. The second part of our contribution involves the simulation of "large" wave propagation problems in acoustics and electromagnetics. In particular, using our multigrid technology along with the adaptive capabilities of the DPG method, we are able to simulate scattering phenomena in the high-frequency regime and solve various problems with singular solutions.

1.5 Outline

This dissertation is organized as follows. Chapter 2 serves as a crash course on the DPG method. The main ideas of the method are described and its three different characterizations are outlined. Necessary notation and definitions for the rest of the document are presented.

Chapter 3 focuses on the application of the DPG method to different variational formulations. As a model problem the two dimensional linear acoustics problem in free space is considered. Results on convergence and the spectrum properties of these formulations are presented. The chapter concludes with numerical experiments focused on the ultraweak formulation in the high-frequency regime, using *hp*-adaptivity. The ability of the DPG method to perform reliable adaptive refinements delivering optimal meshes is showcased by comparisons with an L^2-projection problem.

The next four chapters are devoted to our work on the construction of preconditioners for the DPG method and their application to challenging wave problems. First, in Chapter 4, a theoretical convergence analysis is developed for a one level additive Schwarz preconditioner for the linear acoustics problem. The analysis is based on the subspace correction theory of Xu [117, 118, 119]. The main proof is complemented with the design and execution of a numerical experiment, needed for the computation of an interpolation norm that appears in the theoretical estimates. An extension of the one-level preconditioner to the two-level setting is introduced in Chapter 5. In there, numerical results on the simulation of high-frequency acoustic beams in two space dimensions are displayed. Next, Chapter 6 describes the construction of an adaptive multigrid preconditioner in three space dimensions. Here, the computational complexity is examined in both serial and parallel implementations. Finally, various three dimensional numerical experiments for acoustic and electromagnetic simulations are shown in Chapter 7. The key point, making these simulations computationally tractable, is the integration of the multigrid preconditioner within the adaptive DPG technology. The document concludes with a brief synopsis and a discussion on future directions. Four appendices can be found at the end of the document providing supplemental material for Chapters 2, 4 and 6.

1.6 Acknowledgments

The work presented in this dissertation has been supported with grants by the Air Force Office of Scientific Research (AFOSR grant FA9550-12-1-0484 and FA9550-17-1-0090).

Chapter 2

The discontinuous Petrov–Galerkin method

In this chapter we give a brief overview of the DPG method and highlight some of its important properties. We start by describing the so called *ideal* Petrov–Galerkin method and demonstrate that it can be characterized in three different ways: a) as a minimum residual method, b) as a Petrov–Galerkin method with *optimal test functions* and c) as a mixed formulation. We continue with an overview of the *practical* Petrov–Galerkin method and then we introduce the *discontinuous* Petrov–Galerkin (DPG) method. We conclude the chapter with a discussion on the implementation.

2.1 The ideal Petrov–Galerkin method

Consider the following abstract variational problem. Suppose we are given a continuous bilinear (sesquilinear) form $b(\cdot,\cdot)$ defined on the product $U \times V$ of Hilbert spaces U (trial space) and V (test space), and a continuous linear (anti-linear) form $l(\cdot)$ defined on V. We want to find the solution to the problem:

$$\begin{cases} u \in U \\ b(u,v) = l(v), \quad v \in V. \end{cases} \tag{2.1}$$

We assume that the above continuous problem is well posed, i.e, the continuous inf-sup condition [4] for the bilinear form $b(\cdot,\cdot)$ holds:

$$\gamma = \inf_{u \in U} \sup_{v \in V} \frac{|b(u,v)|}{\|u\|_U \|v\|_V} > 0$$

where γ is the inf - sup constant.

2.1.1 Equivalent characterizations of the ideal PG method

The variational problem (2.1) can be reformulated as an operator equation

$$\mathrm{B}u = l \tag{2.2}$$

where $\mathrm{B} : U \to V'$ is defined as

$$\langle \mathrm{B}u, v \rangle_{V' \times V} = b(u, v)$$

and $\langle \cdot, \cdot \rangle_{V' \times V}$ denotes the duality pairing on $V' \times V$.

Minimum residual formulation. Given a finite-dimensional approximate trial space $U_h \subset U$, (2.2) can be discretized as a minimum residual problem, where we seek an approximate solution $u_h \in U_h$ that minimizes the residual in the norm dual to the test norm, i.e,

$$u_h = \arg\min_{w_h \in U_h} J(w_h), \text{ where } J(w_h) = \frac{1}{2}\|l - \mathrm{B}w_h\|_{V'}^2 \tag{2.3}$$

Unfortunately, computing with the dual norm $\|\cdot\|_{V'}$, is usually not possible. This difficulty can be avoided by introducing the *Riesz operator* for the test space. The Riesz operator

$$\mathrm{R}_V : V \to V'$$

is defined by

$$(\mathrm{R}_V y)(v) = \langle \mathrm{R}_v y, v \rangle_{V' \times V} = (y, v)_V, \quad y, v \in V$$

and it is an isometric isomorphism. This allows us to replace the dual norm $\|\cdot\|_{V'}$ with $\|\mathrm{R}_V^{-1}(\cdot)\|_V$, i.e., (2.3) becomes

$$u_h = \arg\min_{w_h \in U_h} \frac{1}{2}\|\mathrm{R}_V^{-1}(l - \mathrm{B}w_h)\|_V^2. \tag{2.4}$$

From the above minimization problem, we can easily derive two additional formulations. Taking the Gâteaux derivative of (2.4) leads to the following linear problem:

$$\begin{cases} u_h \in U_h \subset U \\ (\mathrm{R}_V^{-1}(l - \mathrm{B}u_h), \mathrm{R}_V^{-1}\mathrm{B}\delta u_h) = 0, \quad \delta u_h \in U_h \end{cases} \tag{2.5}$$

We can now take two different directions. One will lead us to a Petrov–Galerkin method with *optimal test functions* and the second to a mixed method.

Petrov–Galerkin method with optimal test functions. We introduce the *ideal trial-to-test operator* $\mathrm{T} : U_h \to V$, defined by

$$(\mathrm{T}\delta u_h, \delta v)_V = b(\delta u_h, \delta v) \quad \delta u_h \in U_h,\, \delta v \in V, \qquad \text{i.e., } \mathrm{T} = \mathrm{R}_V^{-1}\mathrm{B} \tag{2.6}$$

and we define the *optimal test space* as $V_h^{\mathrm{opt}} := \mathrm{T}(U_h)$, i.e, the image of the trial to test operator T acting on the trial space U_h. Therefore, problem (2.5) leads to the following Petrov–Galerkin scheme:

$$\begin{cases} u_h \in U_h \subset U \\ b(u_h, v_h) = l(v_h), \quad v_h \in V_h^{\mathrm{opt}} \end{cases} \tag{2.7}$$

This particular construction of the test space is crucial because it guarantees uniform and unconditional discrete stability. In general, when the variational problem (2.1) is discretized with an arbitrary choice of trial and test space, discrete stability is not guaranteed even if the problem is stable on the continuous level. In order to ensure discrete stability, the *Babuška's discrete inf-sup condition* [4] has to be satisfied:

$$\gamma_h = \inf_{u_h \in U_h} \sup_{v_h \in V_h} \frac{|b(u_h, v_h)|}{\|u_h\|_U \|v_h\|_V} > 0$$

where γ_h is the *discrete* inf - sup constant. The construction of the test space described above, ensures that the supremum in the above discrete inf sup condition is equal to the supremum over the whole V. Indeed, for $v_h \in V_h^{\mathrm{opt}}$ we have:

$$\frac{b(u_h, v_h)}{\|v_h\|_V} = \frac{b(u_h, \mathrm{T}u_h)}{\|\mathrm{T}u_h\|_V} = \|\mathrm{T}u_h\|_V = \sup_{v \in V} \frac{|(\mathrm{T}u_h, v)_V|}{\|v\|_V} = \sup_{v \in V} \frac{|b(u_h, v)|}{\|v\|_V}. \tag{2.8}$$

Consequently, the Petrov–Galerkin scheme (2.7) inherits automatically the stability from the continuous level with the discrete inf sup constant $\gamma_h \geq \gamma$. It is important to note that the stability is guaranteed even in the pre-asymptotic region. We will demonstrate in the next chapter that this is very beneficial when solving problems with localized solutions, since adaptive refinements can be employed starting from very coarse meshes.

Remark 2.1. It is clear from (2.8) that the Petrov–Galerkin scheme with optimal test functions (2.7) will always deliver a Hermitian positive definite stiffness matrix. We will show how we can exploit this important property later on, when we discuss the construction of a preconditioner for the Conjugate Gradient (CG) method.

Mixed method. Finally, from (2.5) we can identify $\psi := \mathrm{R}_V^{-1}(l - \mathrm{B}u_h)$ to be the *error representation function*, i.e, (2.5) becomes:

$$(\psi, \mathrm{R}_V^{-1}\mathrm{B}\delta u_h) = 0, \quad \delta u_h \in U_h$$

This generates the following mixed problem:

$$\begin{cases} u_h \in U_h,\ \psi \in V, \\ (\psi, v)_V + b(u_h, v) = l(v), \quad v \in V, \\ b(w_h, \psi) \qquad\qquad = 0, \qquad w_h \in U_h \end{cases} \tag{2.9}$$

where now we solve simultaneously for the original unknown u_h and the error representation function ψ. Note that the norm of ψ offers a built-in a-posteriori error indicator. Indeed, if we define the *energy* norm as

$$\|u\|_E := \|\mathrm{B}u\|_{V'} = \sup_{v \in V} \frac{|b(u,v)|}{\|v\|_V}$$

then we can compute the Galerkin error measured in the *energy* norm, by computing the norm of the error representation function (residual).

$$\|u_h - u\|_E = \|\mathrm{B}(u_h - u)\|_{V'} = \|\mathrm{B}u_h - l\|_{V'} = \|\mathrm{R}_V^{-1}(\mathrm{B}u_h - l)\|_V = \|\psi\|_V$$

As we demonstrate later on, the residual is the main tool to drive adaptive refinements in the DPG method.

2.2 The practical Petrov–Galerkin method

The ideal Petrov–Galerkin method promises guaranteed pre-asymptotic stability and a reliable a-posteriori error indicator. These are the main two things one needs in order to build a stable adaptive method. However, the computation of the optimal test space involves the inversion of the Riesz operator R_V, i.e., solution of an additional infinite dimensional boundary value problem (with multiple right hand sides). This is not computationally feasible, and so we consider a *truncated* finite dimensional test space $V^r \subset V$, with $\dim(V^r) >> \dim(U_h)$, the so-called *enriched* test space. Then, the minimum residual formulation (2.3) becomes:

$$u_h = \arg\min_{w_h \in U_h} \frac{1}{2}\|l - \mathrm{B}w_h\|_{V'}^2 = \arg\min_{w_h \in U_h} \frac{1}{2}\|(\mathrm{R}_V^{-1})_r(l - \mathrm{B}w_h)\|_V^2. \tag{2.10}$$

Here, $(\mathrm{R}_V^{-1})_r : V' \to V^r$ is the approximate inverse Riesz operator defined by $(\mathrm{R}_V^{-1})_r l =: v^r$, where

$$\begin{cases} v^r \in V^r \\ (v^r, \delta v)_V = l(\delta v), \quad \delta v \in V^r. \end{cases}$$

The ideal trial-to-test operator (2.6) is now approximated by an approximate operator $\mathrm{T}^r : U_h \to V^r \subset V$ defined by:

$$(\mathrm{T}^r \delta u_h, \delta v)_V = b(\delta u_h, \delta v), \quad \delta u_h \in U_h,\, \delta v \in V^r. \tag{2.11}$$

Equation (2.7) is modified accordingly, and we arrive at a new Petrov–Galerkin scheme for the practical PG method.

$$\begin{cases} u_h \in U_h \subset U \\ b(u_h, v_h) = l(v_h), \quad v_h \in V_r^{\mathrm{opt}} := \mathrm{T}^r(U_h). \end{cases}$$

Finally the mixed formulation (2.9) becomes:

$$\begin{cases} u_h \in U_h,\, \psi^r \in V^r, \\ (\psi^r, v)_V + b(u_h, v) = l(v), \quad v \in V^r, \\ b(w_h, \psi^r) \qquad\qquad = 0, \qquad w_h \in U_h \end{cases}$$

where $\psi^r = (R_V^{-1})_r(l - Bu_h)$ and $\|\psi^r\|_V$ is the approximate residual.

Remark 2.2. Typically, the construction of the enriched test space V^r involves uniform p enrichment over the order of approximation of the trial space. The enrichment order is denoted by Δp, i.e, if the order of the trial functions is p, then the order of the test functions is $r = p + \Delta p$. In all numerical results reported in this work, $\Delta p = 1$ is used.

Remark 2.3. A natural question that one can raise is the following: is discrete stability still guaranteed? There are four studies that we are aware of, attempting to answer this question. First, Gopalakrishnan and Qiu [65], by introducing an appropriate Fortin operator, showed that indeed the approximation maintains discrete stability for two dimensional problems. The construction was later generalized in three dimensions in [19]. Additionally, Carstensen and Hellwig in [20], prove the discrete inf-sup condition for a low order test space for the linear

elasticity problem. Finally, in [98], Nagaraj, Petrides and Demkowicz attempted to quantify the loss of stability by computing norms of relevant Fortin operators, defined on H^1 and $H(\mathrm{div})$ energy spaces in two space dimensions, and showed through numerical experiments that the stability loss is minimal. A brief outline on this work is given in appendix A.

2.3 The *discontinuous* Petrov–Galerkin method. Breaking the test space

So far we outlined, in an abstract way, the main principles of the ideal and the practical PG method and how it can be interpreted in three different ways. Yet, we still need to explain where the word "Discontinuous" is coming from. The approximation of the optimal test functions, naturally adds a significant cost to the overall procedure, since it involves the inversion of the Riesz map on the global level. To avoid this extra cost we "break" the test space, i.e, we test with test functions coming from a *larger broken* test space $V(\Omega_h) \supset V$ that involves no conformity assumptions across inter-element boundaries. This allows for a local inversion of the Riesz map, therefore the additional cost of approximating the optimal test functions is negligible. However, this comes with a price; the introduction of additional interface unknowns (Lagrange multipliers) that live on the whole mesh skeleton. The bilinear form (2.1) is now modified and incorporates these new interface unknowns. The modified problem then reads:

$$\begin{cases} u \in U,\ \hat{u} \in \hat{U} \\ b_h(u,v) + \langle \hat{u}, v \rangle = l(v), \quad v \in V(\Omega_h). \end{cases} \tag{2.12}$$

where $\hat{U}$ is the space of the Lagrange multipliers and $\langle \hat{u}, v \rangle$ denotes an appropriate duality pairing on the mesh skeleton. Note that $b_h(u,v)$ indicates that all differential operators are defined element-wise (see [31]).

Fortunately, the use of broken test spaces does not cause any stability issues. Carstensen et al in [19] show that the resulting variational formulation with broken test spaces remains well posed with a mesh-independent stability constant of the same order as the *inf-sup* constant for the continuous problem.

2.4 Implementation of the DPG method - a short tutorial

We conclude this chapter by providing a short tutorial on how the DPG method can be implemented, and how it can be easily incorporated in any standard FEM code. Additionally, we discuss the discretization of the required energy spaces and lastly we provide the reader with information on the DPG software package used for the numerical simulations presented in this dissertation.

2.4.1 The DPG linear system

Similarly to the derivation of the mixed formulation (2.9) for the practical PG method, we can derive the corresponding mixed formulation for the modified bilinear form (2.12):

$$
\begin{cases}
u_h \in U_h \subset U,\ \hat{u}_h \in \hat{U}_h \subset \hat{U},\ \psi^r \in V^r(\Omega_h), & \\
(\psi^r, v^r)_V + b_h(u_h, v^r) + \langle \hat{u}_h, v^r \rangle = l(v^r), & v^r \in V^r(\Omega_h), \\
b_h(w_h, \psi^r) = 0, & w_h \in U_h, \\
\langle \hat{w}_h, \psi^r \rangle = 0, & \hat{w}_h \in \hat{U}_h,
\end{cases} \tag{2.13}
$$

where $\hat{u}_h$ corresponds to the new interface unknowns that live on the mesh skeleton, and $V^r(\Omega_h)$ denotes the larger broken test space for the practical DPG method.

Let $\mathfrak{U}_\mathfrak{h} = \{\mathfrak{u}_\mathfrak{i}\}_{i=1}^{N}$, $\hat{\mathfrak{U}}_\mathfrak{h} = \{\hat{\mathfrak{u}}_\mathfrak{i}\}_{i=1}^{\hat{N}}$ and $\mathfrak{V}_\mathfrak{r} = \{\mathfrak{v}_\mathfrak{i}\}_{i=1}^{M}$ (where $M > N + \hat{N}$) denote bases for the discrete trial space $U_h \times \hat{U}_h$ and the enriched test space V_r respectively. Then (2.13) can be now reduced to a matrix equation. Indeed, let $\mathsf{B}_{ij} = b(\mathfrak{u}_j, \mathfrak{v}_i)$, $\hat{\mathsf{B}}_{ij} = \langle \hat{\mathfrak{u}}_j, \mathfrak{v}_i \rangle$, $\mathsf{G}_{ij} = (\mathfrak{v}_i, \mathfrak{v}_j)_V$, $\mathsf{l} = l(\mathsf{v}_i)$ and denote by $\mathbb{F}$ to be the field ($\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$). Then, we want to find the set of coefficients $\mathsf{w} = [\mathsf{w}_i]_{i=1}^{N} \in \mathbb{F}^N, \hat{\mathsf{w}} = [\hat{\mathsf{w}}_i]_{i=1}^{\hat{N}} \in \mathbb{F}^{\hat{N}}$ and $\mathsf{q} = [\mathsf{q}_i]_{i=1}^{M} \in \mathbb{F}^M$ such that

$$
u_h = \sum_{i=1}^{N} \mathsf{w}_i \mathfrak{u}_i,\ \hat{u}_h = \sum_{i=1}^{\hat{N}} \hat{\mathsf{w}}_i \hat{\mathfrak{u}}_i \ \text{ and } \psi^r = \sum_{i=1}^{M} \mathsf{q}_i \mathfrak{v}_i,
$$

satisfy

$$
\begin{pmatrix} \mathsf{G} & \mathsf{B} & \hat{\mathsf{B}} \\ \mathsf{B}^* & 0 & 0 \\ \hat{\mathsf{B}}^* & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi^r \\ \mathsf{u}_h \\ \hat{\mathsf{u}}_h \end{pmatrix} = \begin{pmatrix} \mathsf{l} \\ 0 \\ 0 \end{pmatrix}
$$

By solving the first equation for ψ^r and substituting to the next two equations, we obtain the final DPG system:

$$\begin{pmatrix} \mathsf{B}^*\mathsf{G}^{-1}\mathsf{B} & \mathsf{B}^*\mathsf{G}^{-1}\hat{\mathsf{B}} \\ \hat{\mathsf{B}}^*\mathsf{G}^{-1}\mathsf{B} & \hat{\mathsf{B}}^*\mathsf{G}^{-1}\hat{\mathsf{B}} \end{pmatrix} \begin{pmatrix} \mathsf{u}_h \\ \hat{\mathsf{u}}_h \end{pmatrix} = \begin{pmatrix} \mathsf{B}^*\mathsf{G}^{-1}\mathsf{l} \\ \hat{\mathsf{B}}^*\mathsf{G}^{-1}\mathsf{l} \end{pmatrix}. \tag{2.14}$$

Notice that since the test space is discontinuous, the matrix G is block diagonal. This allows us to perform the static condensation of the error representation function ψ^r, *element-wise*. We solve now for u_h and $\hat{\mathsf{u}}_h$ using the standard FEM technology. That is, we compute the matrices of (2.14) for each element, assemble the global stiffness matrix and load vector and solve the linear system as in any standard Galerkin code. Finally, after solving for u_h and $\hat{\mathsf{u}}_h$, we perform local back substitution to compute the element contributions to the error representation function ψ^r, and use their norm as a local error indicator to drive adaptivity. Note that the DPG technology can be implemented within any standard FEM code that supports the discretization of all energy spaces forming the exact sequence, i.e, H^1, $H(\text{curl})$, $H(\text{div})$, L^2, with minimal modifications (mostly for the element matrices computations). We give a more detailed discussion on how the interface unknowns ($\hat{u}_h$) are discretized in the next section.

An alternative approach for solving linear system (2.14) can be exploited by considering an overdetermined linear system. Since matrix G is block diagonal and Hermitian (symmetric), its Cholesky factorization can be utilized in an element-wise fashion. Let $\mathsf{G} = \mathsf{L}\mathsf{L}^*$. Then (2.14) can be rewritten as:

$$\mathsf{A}^*\mathsf{A}\mathsf{x} = \mathsf{A}^*\mathsf{b} \tag{2.15}$$

where

$$\mathsf{A} = [\mathsf{L}^{-1}\mathsf{B} \quad \mathsf{L}^{-1}\hat{\mathsf{B}}], \quad \mathsf{b} = \mathsf{L}^{-1}\mathsf{l} \quad \text{and } \mathsf{x} = [\mathsf{u}_h \quad \hat{\mathsf{u}}_h]^{\mathrm{T}}$$

It is clear that the linear system (2.15) can be interpreted as the normal equations reformulation of the following discrete linear least squares problem

$$\mathsf{A}\mathsf{x} \overset{LS}{=} \mathsf{b} \tag{2.16}$$

where the matrix A is rectangular of size $M \times (N + \hat{N})$. Thus, (2.16) is an overdetermined system and can be solved using specialized linear solvers for least squares problems (e.g QR-based algorithms). Even though, the most common practice in the DPG community is to solve the normal equations, this alternative formulation was studied in [83] and shown to be beneficial in cases where the condition number of the linear system is large. The standard approach of using the normal equations leads to condition number growth of $\mathcal{O}(h^{-2})$ where h is the discretization size. Conversely, in the case of the overdetermined system the condition number would grow only linearly with respect to h^{-1}. We refer the reader to the appendix B for an outline of this work and examples where this alternative approach is preferred.

2.4.2 Discretization - energy spaces and polynomial subspaces

One of the main advantages of the DPG method is its flexibility to be applied in any well posed variational formulation. It is therefore essential to be able to support simultaneous discretization of the whole exact sequence, i.e, H^1, $H(\text{curl})$, $H(\text{div})$, L^2 *energy* spaces.

2.4.2.1 Energy spaces

The standard energy spaces for a bounded domain $\Omega \in \mathbb{R}^3$ are as follows:

$$
\begin{aligned}
L^2(\Omega) &= \{q : \Omega \to \mathbb{R}(\mathbb{C}) : \|q\| < \infty\} \\
H^1(\Omega) &= \{p : \Omega \to \mathbb{R}(\mathbb{C}) : p \in L^2(\Omega),\, \nabla p \in (L^2(\Omega))^3\} \\
H(\text{curl}, \Omega) &= \{E : \Omega \to \mathbb{R}^3(\mathbb{C}^3) : E \in (L^2(\Omega))^3,\, \nabla \times E \in (L^2(\Omega))^3\} \\
H(\text{div}, \Omega) &= \{v : \Omega \to \mathbb{R}^3(\mathbb{C}^3) : v \in (L^2(\Omega))^3,\, \nabla \cdot v \in L^2(\Omega)\}.
\end{aligned}
$$

Note that in the two dimensional case the $\nabla\times$ operator is scalar valued (see Section 2.4.2.2) and the $H(\text{div})$ space is derived by "rotation" of the $H(\text{curl})$ space. Here and for the rest of the document the L^2 norm is denoted by $\|\cdot\|$.

For the broken formulations, we need to introduce the broken test spaces and spaces defined on the mesh skeleton. We assume that the domain Ω is partitioned into open disjoint elements K, and we denote the corresponding mesh by Ω_h. Then, the broken versions of the

standard energy spaces are as follows:

$$
\begin{aligned}
L^2(\Omega_h) &= \{q \in L^2(\Omega) : \forall K \in \Omega_h,\, q|_K \in L^2(K)\} = L^2(\Omega) \\
H^1(\Omega_h) &= \{p \in L^2(\Omega) : \forall K \in \Omega_h,\, p|_K \in H^1(K)\} \supset H^1(\Omega) \\
H(\mathrm{curl}, \Omega_h) &= \{E \in L^2(\Omega)^d : \forall K \in \Omega_h,\, E|_K \in H(\mathrm{curl}, K)\} \supset H(\mathrm{curl}, \Omega) \\
H(\mathrm{div}, \Omega_h) &= \{v \in L^2(\Omega)^d : \forall K \in \Omega_h,\, v|_K \in H(\mathrm{div}, K)\} \supset H(\mathrm{div}, \Omega)
\end{aligned}
$$

For the discretization of the interface unknowns, i.e, the Lagrange multipliers, we need to introduce energy spaces defined on the mesh skeleton. The mesh skeleton is defined by $\Gamma_h = \bigcup_{K \in \Omega_h} \partial K$, where ∂K is the boundary of element K. We additionally assume that the element boundaries ∂K are Lipschitz. We can then define the trace operators on each element as:

$$
\begin{aligned}
H^1(K) \ni p &\longmapsto tr^K_{\mathrm{grad}} p := p|_{\partial K} \in H^{\frac{1}{2}}(\partial K) \\
H(\mathrm{curl}, K) \ni E &\longmapsto tr^K_{\mathrm{curl},\top} E := (n_K \times E) \times n_K|_{\partial K} \in H^{-\frac{1}{2}}(\mathrm{curl}, \partial K) \\
H(\mathrm{curl}, K) \ni E &\longmapsto tr^K_{\mathrm{curl},\perp} E := (n_K \times E)_{\partial K} \in H^{-\frac{1}{2}}(\mathrm{div}, \partial K) \\
H(\mathrm{div}, K) \ni v &\longmapsto tr^K_{\mathrm{div}} v := v|_{\partial K} \cdot n_K \in H^{-\frac{1}{2}}(\partial K)
\end{aligned}
$$

where n_K is the outward unit normal on ∂K. The trace operators tr^K_{grad}, $tr^K_{\mathrm{curl},\top}$, $tr^K_{\mathrm{curl},\perp}$ and tr^K_{div} are defined on each element of the broken spaces given above. Consequently, we can introduce the following mesh trace operators.

$$
\begin{aligned}
H^1(\Omega_h) \ni p &\longmapsto tr_{\mathrm{grad}} p := \prod_{K \in \Omega_h} tr^K_{\mathrm{grad}} p \\
H(\mathrm{curl}, \Omega_h) \ni E &\longmapsto tr_{\mathrm{curl},\top} E := \prod_{K \in \Omega_h} tr^K_{\mathrm{curl},\top} E \\
H(\mathrm{curl}, \Omega_h) \ni E &\longmapsto tr_{\mathrm{curl},\perp} E := \prod_{K \in \Omega_h} tr^K_{\mathrm{curl},\perp} E \\
H(\mathrm{div}, \Omega_h) \ni v &\longmapsto tr_{\mathrm{div}} v := \prod_{K \in \Omega_h} tr^K_{\mathrm{div}} v
\end{aligned}
$$

Finally, the spaces on the mesh skeleton are given by:

$$
\begin{aligned}
H^{\frac{1}{2}}(\Gamma_h) &= tr_{\mathrm{grad}}(H^1(\Omega)) \\
H^{-\frac{1}{2}}(\mathrm{div}, \Gamma_h) &= tr_{\mathrm{curl},\top}(H(\mathrm{curl}, \Omega)) \\
H^{-\frac{1}{2}}(\mathrm{curl}, \Gamma_h) &= tr_{\mathrm{curl},\perp}(H(\mathrm{curl}, \Omega)) \\
H^{-\frac{1}{2}}(\Gamma_h) &= tr_{\mathrm{div}}(H(\mathrm{div}, \Omega))
\end{aligned}
\tag{2.17}
$$

2.4.2.2 Polynomial subspaces

For a simply connected and bounded domain $\Omega \subset \mathbb{R}^d$, the gradient, curl and divergence operators form the following exact sequence in three space dimensions:

$$H^1 \xrightarrow{\nabla} H(\text{curl}) \xrightarrow{\nabla\times} H(\text{div}) \xrightarrow{\nabla\cdot} L^2$$

and the following two exact sequences in two space dimensions:

$$H^1 \xrightarrow{\nabla} H(\text{curl}) \xrightarrow{\nabla\times} L^2$$
$$H^1 \xrightarrow{\text{curl}} H(\text{div}) \xrightarrow{\nabla\cdot} L^2.$$

In the 2D case the $\nabla\times$ and curl operators are defined as follows. For a 2D vector field $E = (E_1, E_2)$, the scalar-valued curl operator is identified as

$$\nabla \times E := \frac{\partial E_2}{\partial x_1} - \frac{\partial E_1}{\partial x_2}.$$

Similarly, for a scalar argument E, the vector-valued curl operator is identified as

$$\text{curl}\, E := \Big(\frac{\partial E}{\partial x_2}, -\frac{\partial E}{\partial x_1}\Big).$$

Recall that a sequence of vector spaces with the corresponding operators is called exact when the range of each operator coincides with the null space of the operator next in the sequence. It is crucial that the exact sequence property is satisfied at the discrete level. That is, the polynomial subspaces

$$W^p \subseteq H^1, \quad Q^p \subseteq H(\text{curl}), \quad V^p \subseteq H(\text{div}), \quad Y^p \subseteq L^2$$

form the analogous exact sequences

$$3D: \quad W^p \xrightarrow{\nabla} Q^p \xrightarrow{\nabla\times} V^p \xrightarrow{\nabla\cdot} Y^p$$
$$2D: \begin{cases} W^p \xrightarrow{\nabla} Q^p \xrightarrow{\nabla\times} Y^p \\ W^p \xrightarrow{\text{curl}} V^p \xrightarrow{\nabla\cdot} Y^p \end{cases}$$

For all the computations of this dissertation, we used the polynomial subspaces described in [53]. In particular, for our numerical experiments we used hexahedral meshes in 3D and

quadrilateral meshes in 2D. The polynomial subspaces for the hexahedron are given by:

$$\begin{aligned} W^p &:= \mathcal{Q}^{p,q,r} \\ Q^p &:= \mathcal{Q}^{p-1,q,r} \times \mathcal{Q}^{p,q-1,r} \times \mathcal{Q}^{p,q,r-1} \\ V^p &:= \mathcal{Q}^{p,q-1,r-1} \times \mathcal{Q}^{p-1,q,r-1} \times \mathcal{Q}^{p-1,q-1} \\ Y^p &:= \mathcal{Q}^{p-1,q-1,r-1} \end{aligned} \tag{2.18}$$

where $\mathcal{Q}^{p,q,r}(x,y,z) = \mathcal{P}^p(x) \otimes \mathcal{P}^q(y) \otimes \mathcal{P}^r(z)$, and $\mathcal{P}^p(x) = \text{span}\{x^j : j = 0, ..., p\}$. These are the standard Nédélec's spaces of the first type. The quadrilateral case is analogous. We note that, throughout the document, we will refer by discretization order, the order corresponding to the exact sequence. For instance a discretization of order 3 would mean that an H^1 variable is discretized with polynomials of order 3 in each direction, but an L^2 variable is discretized with polynomials of order 2.

For the discretization of the trace variables living on the mesh skeleton, the appropriate trace operators defined in the previous section are applied on the above polynomial spaces.

2.4.3 Computer software

All the numerical simulations in this dissertation were implemented in two homegrown software packages, *hp2d* and *hp3d* [25, 37]. These packages are implemented in Fortran 77/90, and they have been developed over the years by Dr. Demkowicz and his students/collaborators. The codes use the recently developed package of orientation embedded high order shape functions for the whole exact sequence for several types of elements described in [53], and they utilize the projection-based interpolation for all energy spaces [38]. As the names suggest, the codes support h and p anisotropic adaptivity with constrained approximation on 1-irregular meshes. Additionally, the codes can accommodate multi-physics problems and curvilinear geometries through transfinite interpolation and isoparametric techniques. Recent additions to the codes, include parallel implementation for shared memory architectures of the element computations, interface with multi-frontal solver packages MUMPS [88, 1] and MKL PARDISO, and implementation of fast integration techniques [95]. Lastly, the two codes include the multigrid solvers, implemented by the author of this dissertation. Their construction was

heavily based on the existing data structures of the two codes and it is described in the following chapters.

At this point we conclude this brief overview of the DPG method and refer the reader to [19, 31, 30, 60] for further reading. In the next chapter we present in detail our work on the DPG method for time harmonic acoustic waves.

Chapter 3

The DPG method for linear acoustics

The problem of interest in this chapter is linear acoustics. First, we demonstrate how different variational formulations can be derived and prove mutual well posedness. Secondly, we present a numerical comparison of the DPG method with the first order least squares (FOSLS) and the standard Galerkin methods in context of their approximability and spectrum properties. We conclude, with an example of a simulation of a high frequency Gaussian beam, where discrete stability and adaptivity play a key role[1].

Author contributions. The contents of this chapter are based on the published paper [106], co-authored by the author of this dissertation. The author of this dissertation contributed to the mathematical theory, software development and numerical simulations related to the work presented in [106].

3.1 Variational formulations

Consider the time harmonic form of the linear acoustics equations derived in Section 1.3.1, for $\Omega \in \mathbb{R}^d$ (with $d = 2, 3$), endowed with homogeneous impedance boundary conditions on the whole boundary

$$\begin{cases} i\omega p + \operatorname{div} u = f_1, & \text{in } \Omega \\ i\omega u + \nabla p \;\; = f_2, & \text{in } \Omega \\ p - u \cdot n \;\;\;\; = 0, & \text{on } \partial\Omega \end{cases} \tag{3.1}$$

[1]The contents of this chapter are partially taken from the published paper: *Petrides, S. and Demkowicz, L. F. (2017). An adaptive DPG method for high frequency time-harmonic wave propagation problems. Comput. Math. Appl., 74(8):1999–2017.*

Depending on with which norm one seeks to measure convergence, the above equations give rise to a total of six different variational formulations, for which it can be shown that they are simultaneously well or ill posed.

Trivial (strong) formulation ($\mathcal{S}$). We start by multiplying the two equations in (3.1) by test functions q and v respectively. We then integrate over the domain Ω and arrive at the *trivial* (or strong) formulation:

$$\begin{cases} (p,u) \in U_{\mathcal{S}} \\ i\omega(p,q) + (\operatorname{div} u, q) = (f_1, q), \quad q \in L^2(\Omega) \\ i\omega(u,v) + (\nabla p, v) \ \ = (f_2, v), \quad v \in (L^2(\Omega))^d \end{cases} \tag{3.2}$$

where

$$U_{\mathcal{S}} = \{(q,v) \in H^1(\Omega) \times H(\operatorname{div}, \Omega) : q - v \cdot n \ = 0 \text{ on } \partial\Omega\} \tag{3.3}$$

and $(f_1, f_2) \in \boldsymbol{L^2} := L^2(\Omega) \times (L^2(\Omega))^d$.

First mixed formulation ($\mathcal{M}_1$). We now integrate by parts the first equation in (3.2) and build the boundary condition into the formulation. We call this process *relaxation*. Then (3.2) produces the mixed formulation:

$$\begin{cases} (p,u) \in U_{\mathcal{M}_1} := H^1(\Omega) \times (L^2(\Omega))^d \\ i\omega(p,q) - (u, \nabla q) + \langle p, q \rangle_{\partial\Omega} = (f_1, q), \quad q \in H^1(\Omega), \\ i\omega(u,v) + (\nabla p, v) \qquad\qquad\quad = (f_2, v), \quad v \in (L^2(\Omega))^d. \end{cases} \tag{3.4}$$

First reduced formulation (Classical) ($\mathcal{R}_1$). Since the second equation in (3.4) is still in the strong form, we can eliminate velocity and obtain a reduced formulation for the pressure or the classical Helmholtz formulation:

$$\begin{cases} p \in U_{\mathcal{R}_1} := H^1(\Omega) \\ (\nabla p, \nabla q) - \omega^2 (p,q) + i\omega \langle p, q \rangle_{\partial\Omega} = (f, q), \quad q \in H^1(\Omega). \end{cases} \tag{3.5}$$

Note that the source term f in (3.5) can now be more irregular. For example for $f_2 \cdot n = 0$ on $\partial\Omega$, we have $(f_2, \nabla q) = -(\operatorname{div} f_2, q)$, where the divergence is understood in the distributional sense. Then in (3.5) $f = i\omega f_1 - \operatorname{div} f_2$.

Second mixed formulation ($\mathcal{M}_2$). Similarly, if we relax only the second equation in (3.2), we have:

$$\begin{cases} (p,u) \in U_{\mathcal{M}_2} := L^2(\Omega) \times V \\ i\omega(p,q) + (\operatorname{div} u, q) = (f_1, q), \quad q \in L^2(\Omega) \\ i\omega(u,v) - (p, \operatorname{div} v) + \langle u \cdot n, v \cdot n \rangle_{\partial\Omega} = (f_2, v), \quad v \in V. \end{cases}$$

Note that the energy space for the velocity incorporates now an extra regularity assumption resulting from building in the impedance boundary condition,

$$V := \{v \in H(\operatorname{div}, \Omega) : v \cdot n \in H^{1/2}(\partial\Omega)\}$$

Second reduced formulation ($\mathcal{R}_2$). Similarly to the classical formulation we can use the first equation to eliminate the pressure. This leads to the standard formulation for the vector Helmholtz equation:

$$\begin{cases} u \in U_{\mathcal{R}_2} := V \\ (\operatorname{div} u, \operatorname{div} v) - \omega^2 (u,v) + \langle u \cdot n, v \cdot n \rangle_{\partial\Omega} = (g, v), \quad v \in V \end{cases}$$

where again g is less regular than L^2. If we assume that $f_2 = 0$ on $\partial\Omega$, then $g = i\omega f_2 - \nabla f_1$, where the gradient is understood in the sense of distributions.

Ultraweak formulation ($\mathcal{U}$). Finally relaxing both equations leads to the *ultraweak* formulation. Note that all derivatives are passed from the trial to the test functions.

$$\begin{cases} (p,u) \in U_{\mathcal{U}} := L^2(\Omega) \times (L^2(\Omega))^d \\ i\omega(p,q) - (u, \nabla q) + i\omega(u,v) - (p, \operatorname{div} v) = (f_1, q) + (f_2, v), \quad (q,v) \in V_{\mathcal{U}} \end{cases} \tag{3.6}$$

where

$$V_{\mathcal{U}} = \{(q,v) \in H^1(\Omega) \times H(\operatorname{div}, \Omega) : q = -v \cdot n \text{ on } \partial\Omega\}$$

3.1.1 Well posedness

In this section we demonstrate that all formulations derived above are simultaneously well or ill posed. We note that in the case of pure Dirichlet boundary conditions, it can be shown that the classical formulation is well posed under the assumption that the frequency does not coincide with a resonance frequency. The proof is based on compact perturbation theory and the Fredholm alternative [101]. The result of the theorem on mutual well posedness is still valid.

We start by stating the *Babuška - Nečas* theorem [4], which is the main tool of proving well-posedness of a variational problem.

Theorem 3.1 (Babuška - Nečas)**.** *Consider the following variational problem:*

$$\begin{cases} u \in U \\ b(u,v) = l(v), \quad v \in V. \end{cases} \tag{3.7}$$

Assume U and V are reflexive spaces, and $b(u,v), l(v)$ are continuous. Additionally let $b(u,v)$ satisfy the inf-sup condition:

$$\gamma = \inf_{u\in U} \sup_{v\in V} \frac{|b(u,v)|}{\|u\|_U \|v\|_V} > 0 \Leftrightarrow \sup_{v\in V} \frac{|b(u,v)|}{\|v\|_V} > \gamma \|u\|_U \tag{3.8}$$

and let $l(v)$ satisfy the compatibility condition:

$$l(v) = 0, \quad v \in V_0 := \{v \in V : b(u,v) = 0, \quad u \in U\}. \tag{3.9}$$

Then the variational problem (3.7) is well-posed, i.e, there exists a unique solution u that depends continuously upon the data:

$$\|u\|_U \leq \frac{1}{\gamma} \|l\|_{V'} = \frac{1}{\gamma} \sup_{v\in V} \frac{|l(v)|}{\|v\|_V}$$

Theorem 3.2. *All the formulations are simultaneously well or ill-posed. That is*

$$\begin{array}{ccc} (\mathcal{R}_1) & \Longleftarrow & (\mathcal{M}_1) \\ \Downarrow & & \Uparrow \\ (\mathcal{S}) & \Longrightarrow & (\mathcal{U}) \\ \Uparrow & & \Downarrow \\ (\mathcal{R}_2) & \Longleftarrow & (\mathcal{M}_2) \end{array}$$

Remark 3.1. Babuška - Nečas theorem is a direct consequence of the *Close Range Theorem* [120, 101]. Indeed, recall the operators generated by the bilinear form, i.e.,

$$\mathrm{B} : U \to V^{'}, \quad \langle \mathrm{B}u, v\rangle_{V^{'}\times V} = b(u,v)$$
$$\mathrm{B}^{'} : V \to U^{'}, \quad \langle \mathrm{B}^{'}v, u\rangle_{U^{'}\times U} = \overline{b(u,v)}$$

Then the inf-sup condition is equivalent to the operator B being bounded below. Additionally, V_0 can be identified by the null space of the transpose operator.

Remark 3.2. In order to prove the bounds, we can always assume the required regularity by first considering $C^{\infty}(\bar{\Omega})$ functions. After the bounds are proven for the smooth functions, we employ density arguments to conclude that they also hold for all (p,u) coming from the domain of the operator. Lastly, we note that the continuity requirement for the bilinear and linear forms can be easily verified, and so it is omitted.

Proof. (Mutual well posedness). We start by the implication $\boldsymbol{(\mathcal{R}_1) \Rightarrow (\mathcal{S})}$. The operator of the strong formulation is defined as:

$$\mathrm{A}(p,u) := (i\omega p + \operatorname{div} u, i\omega u + \nabla p) \tag{3.10}$$

and the domain of A is :

$$D(\mathrm{A}) = \{(q,v) \in H^1(\Omega)\times H(\operatorname{div},\Omega) : q - v\cdot n \ = 0 \text{ on } \partial\Omega\}$$

Since the *energy norm* $\|(p,u)\|_{H^1(\Omega)\times H(\operatorname{div},\Omega)}$ is equivalent to the *graph norm*

$$\|(p,u)\|_G := \left(\|(p,u)\|^2_{\boldsymbol{L^2}} + \|\mathrm{A}(p,u)\|^2_{\boldsymbol{L^2}}\right)^{1/2}$$

it is sufficient to verify the following L^2–bound:

$$\|(p,u)\| \leq C_{\mathrm{S}}\|(f_1,f_2)\| \tag{3.11}$$

The above bound is a direct consequence of Lemmas 4.1 and 4.2 in [33]. The proof is based on the following well known result by Melenk from [90].

Theorem 3.3 (Well posedness of Helmholtz)**.** *Let p such that*

$$\Delta p + \omega^2 p \quad = f, \quad \text{in } \Omega$$

$$\nabla p \cdot n \pm i\omega p = g, \quad \text{on } \partial\Omega$$

with $f \in L^2(\Omega)$ and $g \in H^{1/2}(\partial\Omega)$. For Ω bounded and convex, there exists a $C > 0$ depending only on Ω and an ω_0 such that for $\omega > \omega_0$

$$\|\nabla p\|^2 + \omega^2 \|p\|^2 \leq C(\|f\|^2 + \|g\|^2_{\partial\Omega})$$

The proof of the above theorem is based on Gårding inequality. We note that, in this case the result holds for any value of the frequency. Based on the result above, it can be shown that the stability constants for the strong and ultraweak formulations are independent of the frequency when the graph norm is used instead of the norm induced by the inner product.

Notice that C_{S} in (3.11) can be identified as the L^2 boundedness below constant of the operator A when viewed as a closed operator. It can be easily verified that the operator A is closed, using the definition of distributional derivatives and properties of Sobolev spaces (see [26]). However, if the domain of A is equipped with the graph norm and it is identified as a new energy space, then the operator A is continuous, with continuity constant equal to one. It is then easy to show that the continuous operator is bounded below with a constant of the same order as the L^2 boundedness below constant. Indeed (3.11) implies:

$$\|(p,u)\|^2_{\boldsymbol{L^2}} \leq C_{\mathrm{S}}^2 \|\mathrm{A}(p,u)\|^2_{\boldsymbol{L^2}}.$$

Adding $\|\mathrm{A}(p,u)\|^2$ in both sides and taking the square root gives:

$$\gamma \|(p,u)\|_G \leq \|\mathrm{A}(p,u)\|_{\boldsymbol{L^2}}$$

with

$$\gamma = (1 + C_{\mathrm{S}}^2)^{-\frac{1}{2}}. \tag{3.12}$$

Observe that the constant γ is exactly the inf-sup constant for the bilinear form of the trivial formulation. Indeed for $(p,u) \in U_{\mathrm{S}}$ (defined in (3.3)) and $V := L^2(\Omega) \times (L^2(\Omega))^d$, we have:

$$\sup_{(q,v)\in V} \frac{|b((p,u),(q,v))|}{\|(q,v)\|} = \sup_{(q,v)\in V} \frac{|(\mathrm{A}(p,u),(q,v))|}{\|(q,v)\|} = \|\mathrm{A}(p,u)\|_{\boldsymbol{L^2}} \geq \gamma \|(p,u)\|_G$$

The compatibility condition (3.9) is can be easily verified as follows. First we derive the adjoint operator A^*

$$\begin{aligned}(\mathrm{A}(p,u),(q,v)) &= i\omega(p,q) + (\operatorname{div} u, q) + i\omega(u,v) + (\nabla p, v) \\ &= -(p, i\omega q + \operatorname{div} v) - (u, i\omega v + \nabla q) - \langle u\cdot n, q\rangle_{\partial\Omega} - \langle p, v\cdot n\rangle_{\partial\Omega}\end{aligned}$$

The boundary terms are eliminated if $q = -v\cdot n$ on $\partial\Omega$. Therefore the adjoint is defined as:

$$\mathrm{A}^*(q,v) = -(i\omega q + \operatorname{div} v, i\omega v + \nabla q) = -\mathrm{A}(q,v)$$

and its domain is:

$$D(\mathrm{A}^*) = \{(q,v) \in H^1(\Omega)\times H(\operatorname{div},\Omega) : q + v\cdot n \ = 0 \text{ on } \partial\Omega\}$$

Operator A is thus skew-adjoint. Its null space is given by the solution of the system below.

$$\begin{aligned} i\omega q + \operatorname{div} v &= 0 \\ i\omega v + \nabla q &= 0 \end{aligned}$$

Eliminating v gives:

$$-\Delta q - \omega^2 q = 0$$

Assuming regularity, q satisfies the classical variational formulation, which is well posed by assumption. This means that q admits only the trivial solution, i.e, $q = 0$. Consequently, the null space of the adjoint is trivial and this concludes the proof of $(\mathcal{R}_1) \Rightarrow (\mathcal{S})$.

$(\mathcal{S}) \Rightarrow (\mathcal{U})$. The proof of this implication is a direct consequence of the Close Range Theorem for both closed and continuous operators. Indeed, the assumption that the strong formulation is well posed implies

$$C\|(p,u)\| \le \|\mathrm{A}(p,u)\|, \qquad \forall (p,u)\in D(\mathrm{A}) \tag{3.13}$$

This bound shows that the range of A is closed. Therefore by the Closed Range Theorem for closed operators the range of A^* is also closed. Combined with injectivity then the following bound for A^* is true

$$C\|(q,v)\| \le \|\mathrm{A}^*(q,v)\|, \qquad \forall (q,v)\in D(\mathrm{A}^*) \tag{3.14}$$

Notice that the constant C is the same for both bounds (3.13) and (3.14). Additionally, the bilinear form of the ultraweak formulation $(u, \mathrm{A}^* v)$ generates the operator $\mathrm{B} : U \to V^{'}$, where $U = L^2(\Omega) \times (L^2(\Omega))^d$ and $V = D(\mathrm{A}^*)$. Its transpose $\mathrm{B}^{'} : V \to U^{'}$ is the operator corresponding to the trivial formulation for the adjoint operator, which is assumed to be well posed. Therefore, by the Close Range Theorem for Continuous operators we conclude that the inf-sup constant for the ultraweak formulation equals to the inf-sup constant of the operator $\mathrm{B}^{'}$. This inf-sup constant is now related to the L^2 boundedness below constant of A^* through (3.12). Since, the L^2 boundedness below constants for operator A and A^* are equal, we conclude that the inf-sup constants for the trivial and the ultraweak formulations are identical. Note that the compatibility condition follows from injectivity of A.

$(\mathcal{U}) \Rightarrow (\mathcal{M}_1)$. Denote by $\mathfrak{u} = (p, u)$ and $\mathfrak{v} = (q, v)$ the group trial and trace variable respectively. Then, we need to verify the assumption (3.8) for

$$b(\mathfrak{u}, \mathfrak{v}) = b^{\mathcal{M}_1}(\mathfrak{u}, \mathfrak{v}) := i\omega(p, q) - (u, \nabla q) + i\omega(u, v) + (\nabla p, v) + \langle p, q \rangle_{\partial\Omega}$$

and $U = U_{\mathcal{M}_1} = V = V_{\mathcal{M}_1} = H^1(\Omega) \times (L^2(\Omega))^d$. Integrating $(\nabla p, q)$ by parts and applying the boundary condition we arrive at the ultraweak bilinear form

$$b^{\mathcal{U}}(\mathfrak{u}, \mathfrak{v}) := i\omega(p, q) - (u, \nabla q) + i\omega(u, v) - (p, \operatorname{div} v)$$

where $U = U_{\mathcal{U}}$ and $V = V_{\mathcal{U}}$ as in (3.6). Then for the mixed formulation we have:

$$\sup_{\mathfrak{v} \in V_{\mathcal{M}_1}} \frac{|b^{\mathcal{M}_1}(\mathfrak{u}, \mathfrak{v})|}{\|\mathfrak{v}\|_{V_{\mathcal{M}_1}}} \geq \sup_{\mathfrak{v} \in V_{\mathcal{U}}} \frac{|b^{\mathcal{M}_1}(\mathfrak{u}, \mathfrak{v})|}{\|\mathfrak{v}\|_{V_{\mathcal{U}}}} = \sup_{\mathfrak{v} \in V_{\mathcal{U}}} \frac{|b^{\mathcal{U}}(\mathfrak{u}, \mathfrak{v})|}{\|\mathfrak{v}\|_{V_{\mathcal{U}}}} \geq C\|\mathfrak{u}\|_{\boldsymbol{L^2}}. \tag{3.15}$$

where the norms on the tests spaces are given by

$$\|(q, v)\|_{V_{\mathcal{U}}} := \|(q, v)\|_{H^1(\Omega) \times H(\operatorname{div}, \Omega)}$$

$$\|(q, v)\|_{V_{\mathcal{M}_1}} := \|(q, v)\|_{H^1(\Omega) \times (L^2(\Omega))^d}.$$

Note that the first inequality is a consequence of the fact that $V_{\mathcal{U}}$ is contained in $V_{\mathcal{M}_1}$ and the last from well posedness of the ultraweak formulation. It remains to bound $\|\nabla p\|$. Using the second equation in (3.4) we have

$$\|\nabla p\|^2 = \|f_2 - i\omega\|^2 \leq 2(\|f_2\|^2 + \omega^2 \|u\|^2)$$

Combining the above with the L^2-bound (3.15) the result follows. The compatibility condition is a direct consequence of integration by parts and well posedness of the ultraweak formulation.

$(\mathcal{M}_1) \Rightarrow (\mathcal{R}_1)$. By introducing a new variable $u = -\nabla p/i\omega$ in the reduced formulation (3.5), we can easily see that (p, u) satisfies the first mixed formulation (3.4). Since the mixed formulation is well posed, then the following bound holds:

$$\|p\|^2_{H^1(\Omega)} + \|u\|^2 \leq C\|f\|^2_{(H^1(\Omega))'}$$

and this proves boundedness below. The compatibility condition can be verified with similar reasoning as before. We conclude the proof of the theorem by noting that the proofs of the rest of the implications are similar to the ones presented. □

3.1.2 Broken formulations

We are now ready to derive the corresponding "broken formulations". The test spaces are now discontinuous, i.e., there is no global conformity requirement. However, additional interface unknowns (Lagrange multipliers) are introduced, which live on the whole mesh skeleton Γ_h. The procedure is as follows. We multiply the original problem by broken test functions and we consider the element bilinear forms, where the integration is done over the element. Finally, to get the global bilinear form, we take the sum over all elements. If we recall the main result of [19], we can conclude that all DPG "broken" formulations are also simultaneously well posed, provided that the original continuous problem is well posed. In fact the stability constant is independent of the mesh and it is of the same order of the original inf-sup constant.

Broken trivial formulation. In the case of the trivial formulation, L^2-conformity does not imply any continuity conditions between elements. Therefore, the corresponding broken formulation for the trivial formulation is identical to (3.2). Testing with optimal test functions leads to the usual first order least squares method (FOSLS) [11, 17].

Broken primal formulation. The broken primal formulation is obtained by breaking the classical formulation (3.5). The formulation reads:

$$\begin{cases} p \in H^1(\Omega), \hat{u}_n \in H^{-\frac{1}{2}}(\Gamma_h) \\ p - \hat{u}_n = 0, \text{ on } \partial\Omega \\ (\nabla p, \nabla_h q) - \omega^2 (p, q) + \langle \hat{u}_n, q \rangle_{\Gamma_h} = (f, q), \quad q \in H^1(\Omega_h) \end{cases}$$

The test function q now comes from the "broken" H^1 space, denoted by $H^1(\Omega_h)$. The symbol ∇_h denotes an element-wise operation and $\langle \cdot, \cdot \rangle_{\Gamma_h}$ denotes the appropriate duality pairing on the mesh skeleton Γ_h.

Broken ultraweak formulation. Similarly with the primal formulation, we multiply (3.2) by broken test functions u, v and integrate by parts element-wise. We finally get:

$$\begin{cases} p \in L^2(\Omega), \hat{p} \in H^{\frac{1}{2}}(\Gamma_h) \\ u \in (L^2(\Omega))^d, \hat{u}_n \in H^{-\frac{1}{2}}(\Gamma_h) \\ \hat{p} - \hat{u}_n = 0, \text{ on } \partial\Omega \\ i\omega (p, q) - (u, \nabla_h q) + \langle \hat{u}_n, q \rangle_{\Gamma_h} \qquad = (f_1, q), \quad q \in H^1(\Omega_h) \\ i\omega (u, v) - (p, \operatorname{div}_h v) + \langle \hat{p}, v \cdot n \rangle_{\Gamma_h} = (f_2, v), \quad v \in H(\operatorname{div}, \Omega_h) \end{cases}$$

Again, $H^1(\Omega_h)$ and $H(\operatorname{div}, \Omega_h)$ denote the broken test spaces and div_h and ∇_h denote element wise operations.

3.1.3 The ultraweak formulation

Convergence of the DPG method for the Helmholtz equation was extensively studied in [122], where Zitteli et al. proved that the ultraweak DPG method is pollution free in the one dimensional case. Moreover, Demkowicz et al. in [33] studied the multidimensional Helmholtz equation and proved theoretical convergence rates, that explicitly show the dependence on the frequency, the mesh size and the order of approximation. We note that both studies where focused on the ultraweak formulation of the DPG method. Although, their study covered the

case of the impedance boundary condition, their main result on error estimates is valid also for other boundary conditions. Their main results are outlined below.

Denote by $\mathfrak{u} = (p, u)$ the group unknown, $\mathfrak{v} = (q, v)$ the group test function and $\hat{\mathfrak{u}} = (\hat{p}, \hat{u}_n)$ the group trace. Consider the operator A of (3.10), and choose the *adjoint* norm for the test space, i.e, $\|\mathfrak{v}\|_V = \|\mathrm{A}^*\mathfrak{v}\|$. Then the ideal PG method with optimal test functions delivers L^2 projection. We can easily verify this result by the following reasoning. It is well known [122] that the Petrov–Galerkin method with optimal test functions (2.7) delivers an orthogonal projection in the *energy* norm $\|\cdot\|_E$ defined by:

$$\|\mathfrak{u}\|_E = \sup_{\mathfrak{v}\in V} \frac{|b(\mathfrak{u}, \mathfrak{v})|}{\|\mathfrak{v}\|_V}.$$

Therefore, in the case of the ultraweak formulation the energy norm coincides with the original L^2 norm in V:

$$\|\mathfrak{u}\|_E = \sup_{\mathfrak{v}\in V} \frac{|b(\mathfrak{u}, \mathfrak{v})|}{\|\mathfrak{v}\|_V} = \sup_{\mathfrak{v}\in V} \frac{|(\mathfrak{u}, \mathrm{A}^*\mathfrak{v})|}{\|\mathrm{A}^*\mathfrak{v}\|} = \|\mathfrak{u}\|.$$

However for the broken formulation, the optimal test norm is not *localizable*, i.e, it stops being a norm for the larger, broken test space. A quasi-optimal test norm is introduced [122] by augmenting the adjoint norm with an extra L^2 term, i.e,

$$\|\mathfrak{v}\|_V^2 = \|\mathrm{A}^*\mathfrak{v}\|^2 + \alpha\|\mathfrak{v}\|^2, \quad \text{where} \quad \alpha = \mathcal{O}(1). \tag{3.16}$$

We refer to this norm as the *adjoint graph norm*. The effect of scaling parameter α is studied in detail in [62] using dispersion analysis, wherein Gopalakrishnan et al. show that as the parameter approaches zero, the DPG method delivers an orthogonal projection (in a specific frequency-dependent norm) for the traces. Under certain circumstances this produces "qualitatively better" results for the field variables.

Recall that for the problem of interest, both A and A^* are bounded below with the same frequency independent constant, i.e,

$$\|\mathrm{A}\mathfrak{u}\| \geq \gamma\|\mathfrak{u}\|, \quad \|\mathrm{A}^*\mathfrak{v}\| \geq \gamma\|\mathfrak{v}\|.$$

Consequently, the original and modified test norms are robustly equivalent. Indeed,

$$\|\mathrm{A}^*\mathfrak{v}\|^2 \leq \|\mathfrak{v}\|_V^2 \text{ and } \|\mathfrak{v}\|_V^2 \leq (1 + \frac{\alpha}{\gamma^2})\|\mathrm{A}^*\mathfrak{v}\|^2.$$

Recalling the main result of the paper of Carstensen et al. [19], it can be concluded that, the robust stability constant is maintained for the broken spaces. This in turn implies the following estimate proven in [33], i.e., the approximation error of the ultraweak DPG formulation is bounded by the corresponding best approximation error uniformly in frequency ω.

$$\|\mathfrak{u}-\mathfrak{u}_h\|^2+\|\hat{\mathfrak{u}}-\hat{\mathfrak{u}}_h\|_Q^2 \leq C\Big[\inf_{w_h}\|\mathfrak{u}-w_h\|^2+\inf_{\hat{w}_h}\|\hat{\mathfrak{u}}-\hat{w}_h\|_Q^2\Big] \tag{3.17}$$

Note that, the error in the group trace variable $\hat{\mathfrak{u}}$ is measured in a special minimum energy extension norm $\|\cdot\|_Q$ defined by

$$\|(\hat{p},\hat{u}_n)\|_Q=\inf\{\|(p,u)\|_G : \forall (p,u)\in H_A \text{ such that } \mathrm{tr}_{\Gamma_h}(p,u)=(\hat{p},\hat{u}_n)\}$$

where

$$H_A:=\{(p,u)\in H^1(\Omega)\times H(\mathrm{div},\Omega) : p-u\cdot n=0 \quad \text{on } \partial\Omega\}.$$

It is important to note that the constant C in the error estimate (3.17) is independent of the frequency ω. The second term of the right hand side in (3.17), represents the norm of the best approximation error in the interface unknowns.

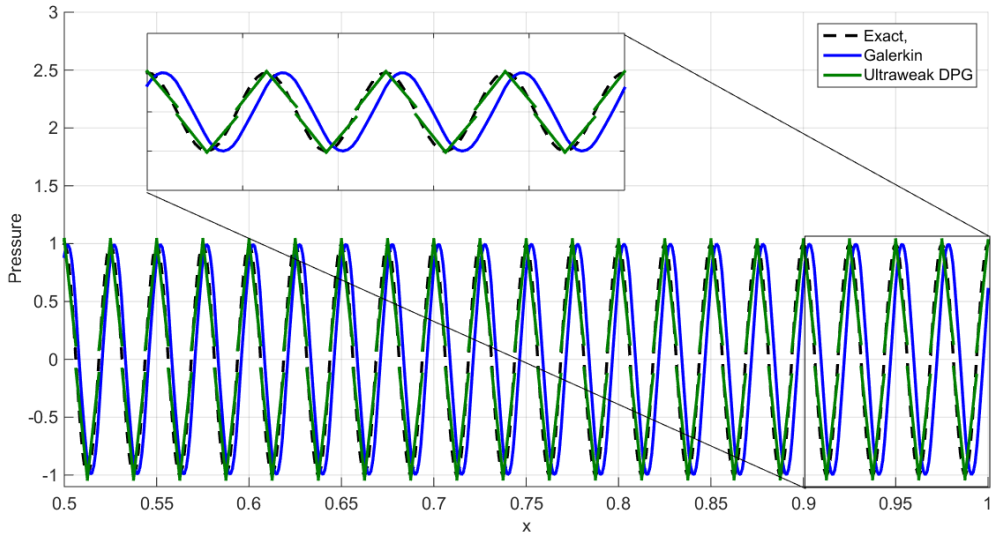


Figure 3.1: Standard Galerkin vs Ultraweak DPG for four quadratic elements per wavelength. In 1D, contrary to the standard Galerkin method the ultraweak DPG is pollution free.

In the special case of one space dimension this term is zero, since the traces are in fact just numbers at grid points. Moreover, the first term of the right hand side in (3.17) represents the L^2 projection error in the field unknowns and does not suffer from pollution. This gives an upper bound for the total error that depends only on the product ωh and the order of approximation p, i.e., there is no extra ω term. Therefore, the ultraweak formulation in one space dimension is a pollution free method (see Figure 3.1). This important result is discussed in detail in [122]. Unfortunately, this is not the case in the multidimensional case. Even if the constant C is independent of ω, the pollution arises from the best approximation error of the interface unknowns, i.e, the $\|\cdot\|_Q$ is *not* pollution free. The final estimate from [33] for tetrahedral meshes for the lowest order of approximation reads:

$$\|(\mathfrak{u},\hat{\mathfrak{u}}) - (\mathfrak{u}_h,\hat{\mathfrak{u}}_h)\|_U \leq Ch\omega^2$$

This estimate shows that even if ωh is kept fixed, the numerical approximation is "polluted" as the frequency grows. An additional estimate is proven in [33], that shows that pollution can be controlled using high order approximations.

3.2 Numerical results

In this section we show some numerical results for different formulations of problem (3.1). First, we examine the convergence for the trivial (FOSLS), the DPG primal and the DPG ultraweak formulations and compare them with the Bubnov-Galerkin method based on the first reduced (classical) variational formulation. Next, we study the spectrum properties of the stiffness matrix of the DPG formulations.

For the experiment, we consider a square domain $\Omega = (0,1)^2$, and we use a Gaussian beam [105] as a manufactured solution. The formula for the Gaussian beam is given by:

$$p(x,y,z) = p_0 \frac{w_0}{w(z)} e^{-\frac{r^2}{w^2(z)}}\, e^{-i(kz + k\frac{r^2}{2R(z)} - \phi(z))}$$

where r is the radial distance from the center axis of the beam, z is the axial distance from the beam's focus, $k = \frac{2\pi}{\lambda}$ is the wavenumber, $w(z) = w_0\sqrt{1+(z/z_R)^2}$ is the spot size, $R(z) =$

$z(1+(z_R/z)^2)$ is the radius of curvature, and $\phi(z) = \tan^{-1}(z/z_R)$ is *Gouy phase shift* at z. Additionally, we use inhomogeneous impedance boundary conditions, with the impedance data computed by lifting the boundary data of the manufactured solution into the finite element space.

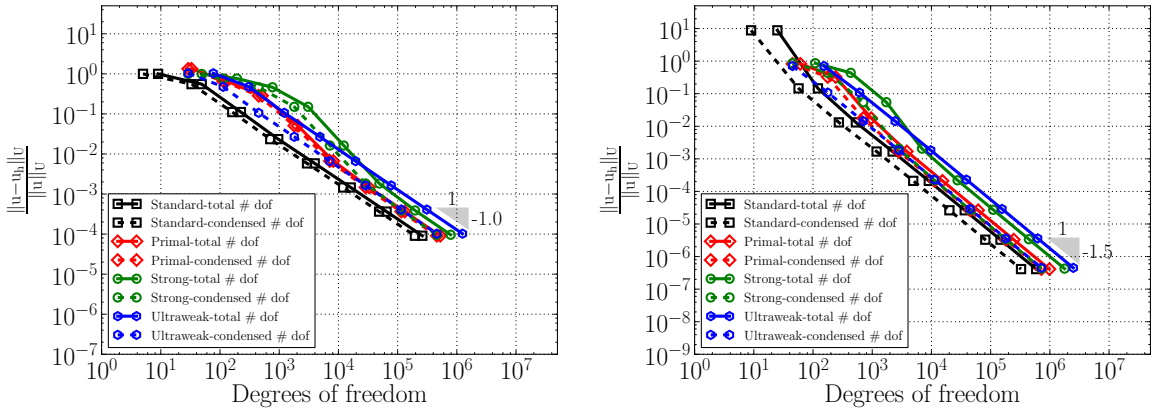
3.2.1 Convergence rates

For the convergence rates, we perform successive uniform $h-$refinements, starting from an initial uniform mesh of four squares. Since the exact solution is smooth, the expected asymptotic rate of convergence is h^p [33], where p is the order of approximation of the trial space corresponding to the exact sequence and h is the size of the side of a square element. In terms of the number of degrees of freedom (N) the expected asymptotic rate of convergence is $N^{-\frac{p}{d}}$, where d is the dimension of the problem. In our case $d = 2$, so the expected asymptotic rate of convergence is $N^{-\frac{p}{2}}$.

In Figures 3.2 and 3.3 we show results for frequency $\omega = 4.6\pi$. In Figure 3.2 we present error convergence rates for $p = 2, 3, 4, 5$. The y-axis represents the relative field error in the appropriate trial norm $\|\cdot\|_U$, i.e, $U = H^1(\Omega)$ for the standard Galerkin and the DPG primal formulations, $U = H^1(\Omega) \times H(\text{div}, \Omega)$ for the trivial and $U = L^2(\Omega) \times (L^2(\Omega))^2$ for the DPG ultraweak formulation. We plot the relative error against both the size of the total DPG system and the condensed DPG system. The condensed DPG system is obtained after eliminating all the interior degrees of freedom, i.e, the degrees of freedom associated with the middle node of an element. It is important to note that in the case of the ultraweak formulation the trial variables u and p are discontinuous. Therefore, they can be all condensed out of the final system, which now contains only the degrees of freedom of the interface unknowns $\hat{u}_n$ and $\hat{p}$. Additionally, we note that this elimination is performed on the element level, and therefore is computationally inexpensive.

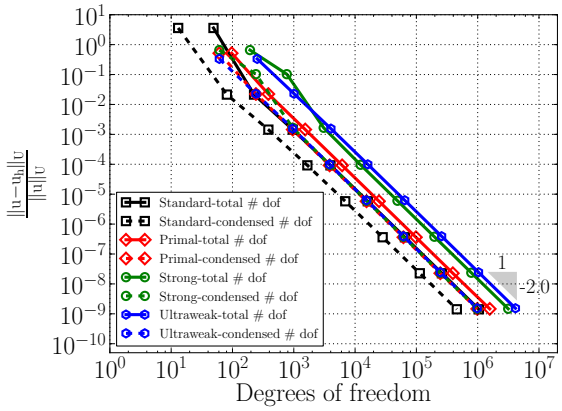
We have the following observations. First, all formulations recover the theoretical convergence rates. Secondly, all DPG formulations give the same relative error in the asymptotic regime for the same mesh. In fact, we can see that the condensed DPG system has exactly the

same size for all DPG formulations. Unfortunately, the DPG system is approximately twice as large as the standard Galerkin system, since we have to solve for the additional interface unknowns. Finally, notice that in the pre-asymptotic region the DPG ultraweak seems superior with respect to the other DPG formulations. This is somewhat expected since, the convergence in the ultraweak formulation is only in the L^2 sense. On the other hand for the trivial and DPG primal formulation the convergence is in H^1 and $H(\text{div})$ norms, which contain derivatives.

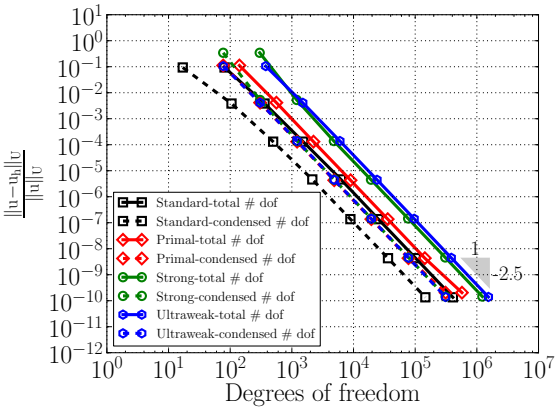


(a) $p = 2$

(b) $p = 3$



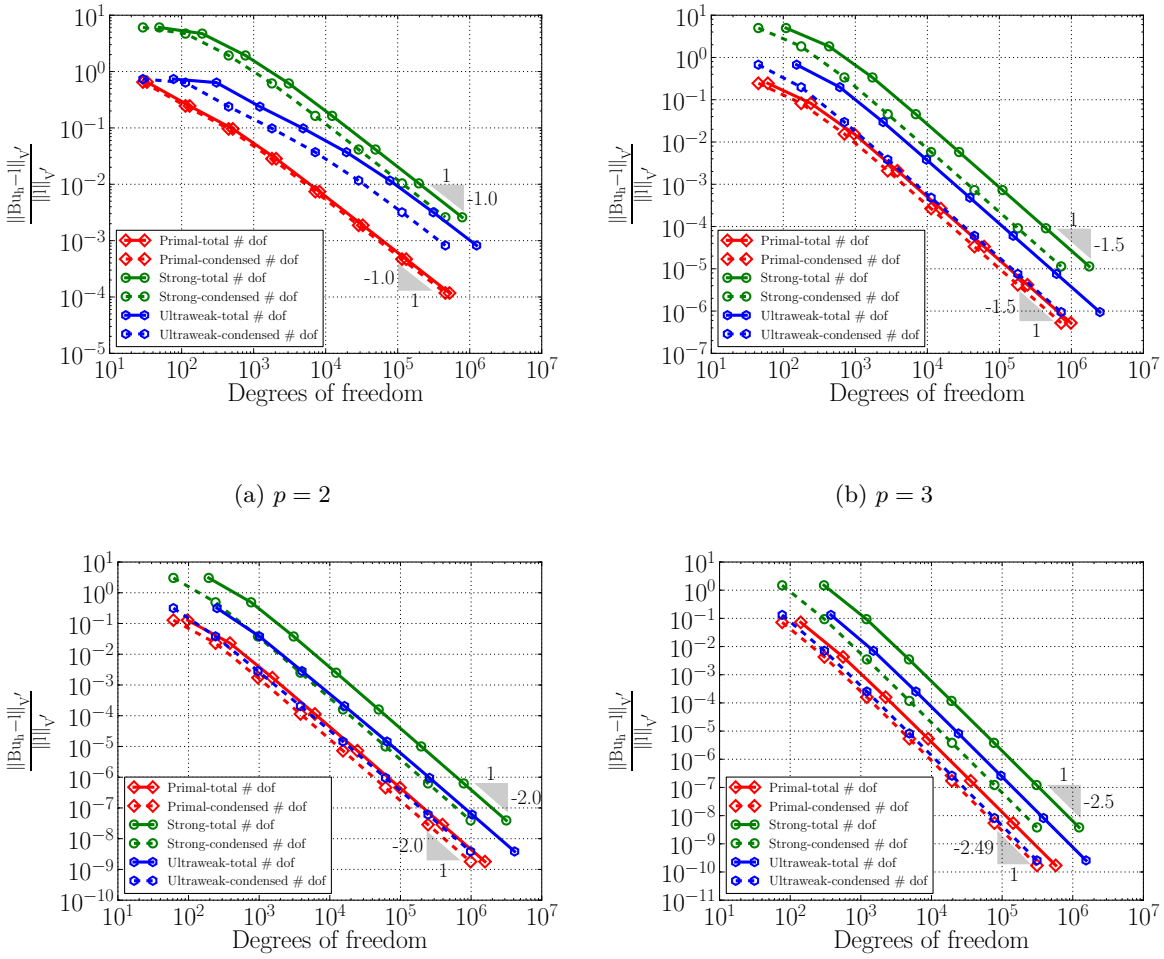
(c) $p = 4$



(d) $p = 5$

Figure 3.2: Relative error convergence rates for the linear acoustics problem in 2D with the a Gaussian beam of frequency $\omega = 4.6\pi$ as a manufactured solution. The expected convergence rate (h^p or $N^{-p/2}$) is recovered

In Figure 3.3 we show convergence rates of the residuals. Note that the standard Galerkin formulation is not included since, contrary to minimum residual methods like DPG, we have no direct access to the residual. In order to have a dimensionless quantity to compare all the formulations we use relative residual with respect to the norm of the load ($\|l\|_{V'}$). As expected all formulations recover the theoretical rates.



(a) $p = 2$

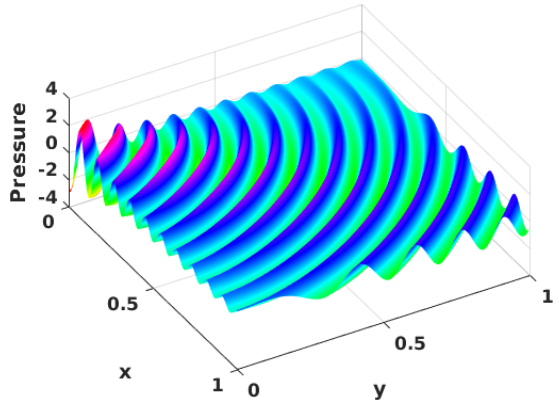
(b) $p = 3$

(c) $p = 4$

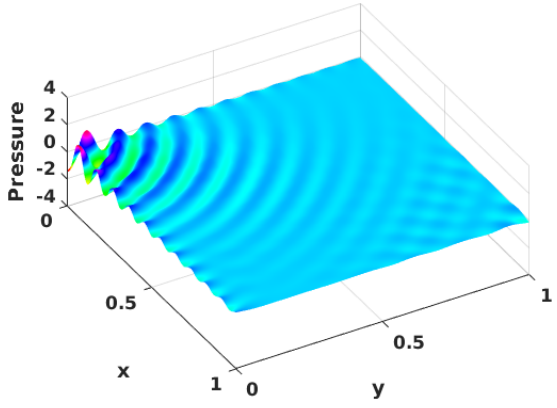
(d) $p = 5$

Figure 3.3: Relative DPG Residual convergence rates for the linear acoustics problem in 2D with the a Gaussian beam of frequency $\omega = 4.6\pi$ as a manufactured solution. The expected convergence rate (h^p or $N^{-p/2}$) is recovered.

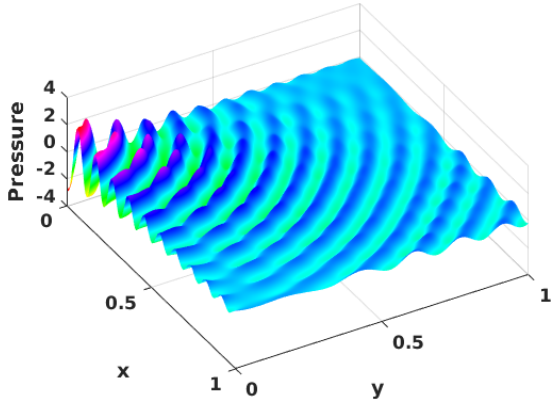
In order to get a better idea of how the different formulations perform we need to visualize the numerical solutions. In Figure 3.4 we show the exact and the computed pressure for all formulations. These solutions are computed with a uniform mesh of 400 cubic elements. The frequency is 20π (i.e., approximately 15 wavelengths in the diagonal direction with 45° degrees angle). It is clear that the trivial and the DPG primal formulation, give more diffusive solutions. On the other hand, the standard Galerkin and the ultraweak formulation are very close to the exact solution.



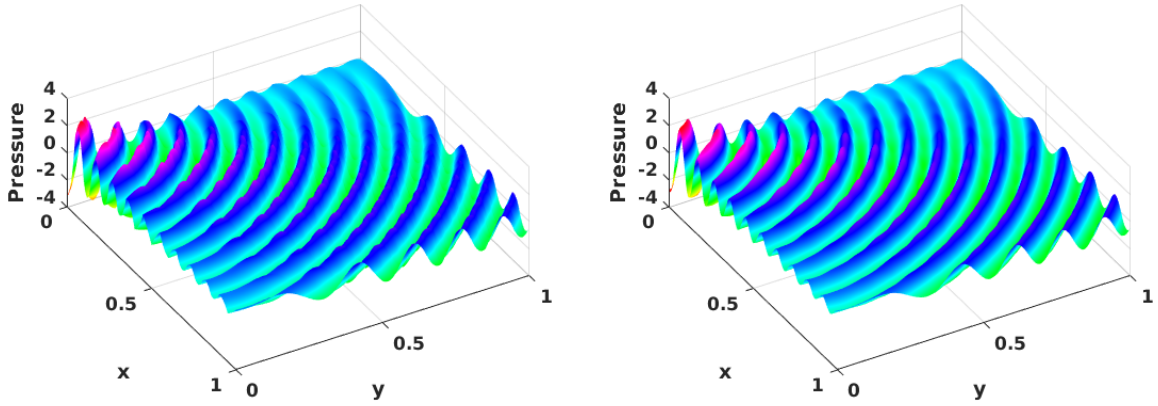
(a) Exact solution



(b) FOSLS



(c) DPG Primal



(d) DPG ultraweak

(e) Standard Galerkin

Figure 3.4: Real part of the exact and numerical pressure for all formulations. Simulation of a Gaussian beam of frequency $\omega = 20\pi$ on a uniform mesh of 400 square elements of polynomial order $p = 3$. Notice that FOSLS and DPG primal formulations are very diffusive.

A similar behavior can be observed in one dimensional implementation, shown in Figure 3.5. In this case we use the manufactured solution $u_{exact} = e^{-i\omega x}$, with a soft boundary condition at $x = 0$ and an impedance boundary condition at $x = 1$. The frequency is $\omega = 20\pi$ and the mesh consists of 50 quadratic elements.

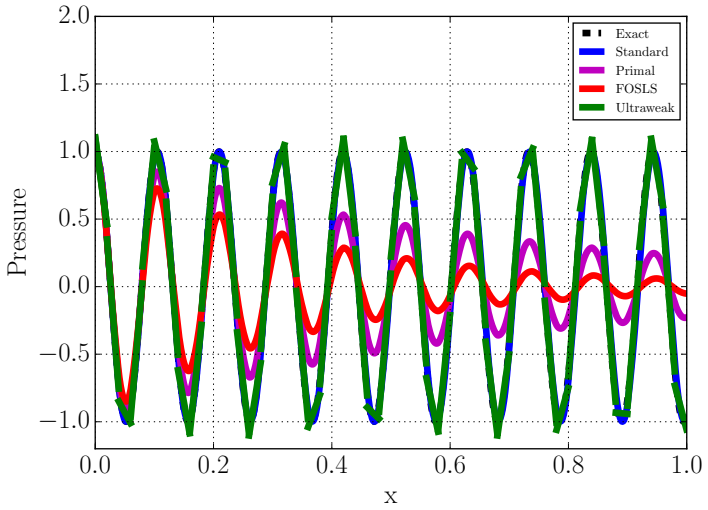


Figure 3.5: Real part of the exact and numerical pressure for all formulations 1D. The FOSLS and the DPG primal formulations deliver very diffusive solutions.

3.3 Conditioning Study

Numerical methods are developed on computers which have finite precision, i.e, a number is represented by a fixed number of significant digits, and so at some point rounding must occur. This gives rise to the so called *roundoff* error and unfortunately it is something that can not be avoided. Roundoff error causes small perturbations in both left and right hand sides of the linear system and this leads to a perturbed solution. The condition number of a matrix A, $\kappa(\mathsf{A})$, is a measure of how sensitive the solution is to the roundoff error. The definition of the condition number is given by

$$\kappa(\mathsf{A}) = \|\mathsf{A}\|\|\mathsf{A}^{-1}\|$$

where $\|\cdot\|$ can be any matrix norm. If we choose the 2-norm $\|\mathsf{A}\|_2 := \max_{\mathsf{x}\neq 0} \frac{\|\mathsf{A}\mathsf{x}\|_2}{\|\mathsf{x}\|_2}$, we arrive to the *spectral* condition number for a general matrix A:

$$\kappa(\mathsf{A}) = \frac{\sigma_{max}}{\sigma_{min}}$$

where σ_{max} and σ_{min} denote the maximum and minimum singular values of matrix A respectively. For the particular case of the DPG method, the stiffness matrix is always Hermitian (or symmetric) and therefore, the condition number can be computed using the eigenvalues of the matrix itself, i.e,

$$\kappa(\mathsf{A}^{\mathsf{DPG}}) = \frac{\lambda_{max}}{\lambda_{min}}$$

where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of matrix $\mathsf{A}^{\mathsf{DPG}}$ respectively.

The condition number of a matrix affects the behavior of linear solvers, both direct and iterative ones. In particular, it appears in the convergence estimates for the iterative solvers. For the DPG method, the Conjugate Gradient algorithm is of great interest, since it is the best candidate for Hermitian positive definite systems. Direct solvers are also affected as the condition number grows. As a rule of thumb, m digits of accuracy may be lost, if the condition number $\kappa(\mathsf{A}) = \mathcal{O}(10^m)$.

3.3.1 Results on conditioning

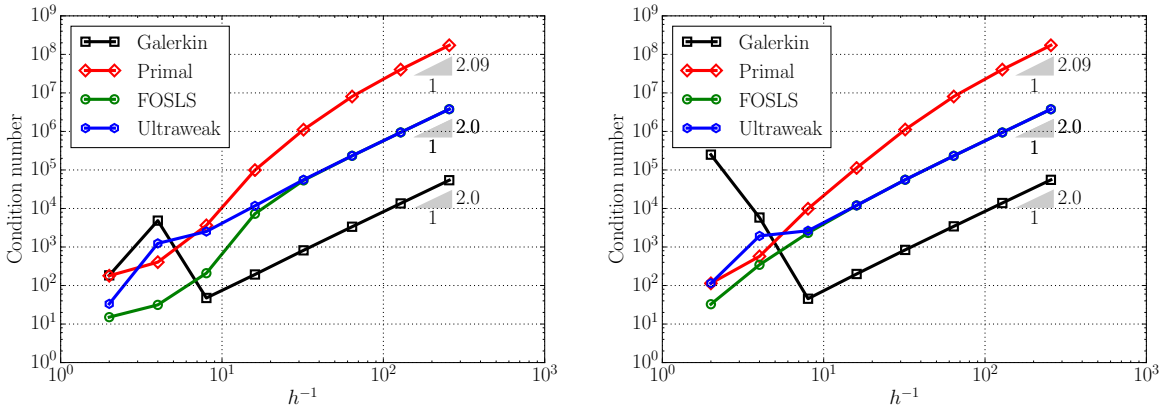
In this section, we study the conditioning of the final DPG system for all the formulations considered in Section 3.2.1. In particular, we are interested in the condition number of the global stiffness matrix after the static condensation of the interior degrees of freedom, since this is the matrix that eventually goes into the solver. Additionally, we apply diagonal scaling on the matrix before computing its condition number. Examining the diagonally scaled matrix seems to be suitable for two reasons: a) it is a computationally inexpensive procedure but it significantly improves the condition number, especially if the matrix is diagonally dominant, and b) such a scaling is applied explicitly in many preconditioning techniques for iterative solvers and implicitly in direct solvers by performing pivoting.

We note that for the computation of the minimum and maximum eigenvalue, we employ the singular value decomposition (SVD) algorithm using the LAPACK package, unless the size of the matrix is too large. In such a case we exploit the sparsity of the matrix and apply power iteration techniques.

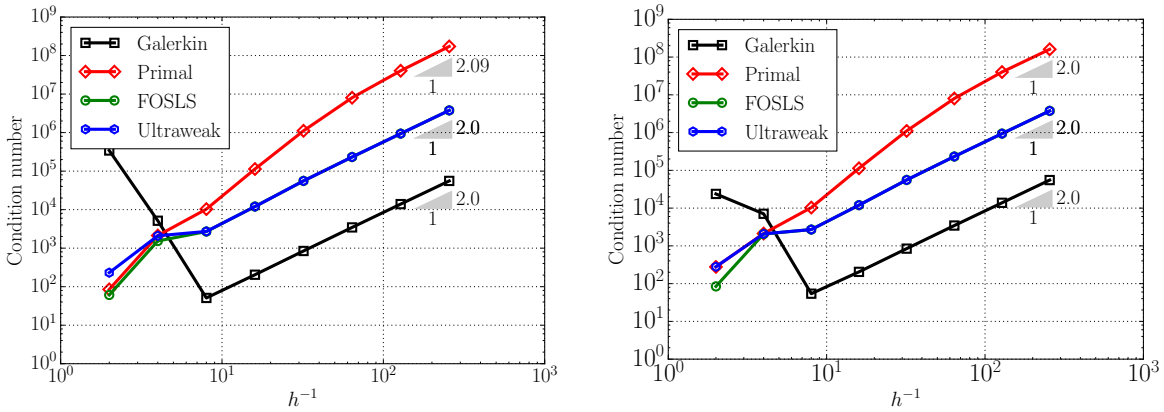
We consider the same problem as in Section 3.2.1. We start with a uniform mesh of four elements and study how the condition number grows as we successively perform several uniform h-refinements. Again, we consider the cases where the order of approximation $p = 2, 3, 4, 5$. The results on the condition number for the condensed matrix are reported in Figure 3.6. As it can be verified from the graph, for all formulations the condition number grows quadratically with respect to h^{-1}. This confirms the theoretical proof in [65], where the authors show that the condition number is $\mathcal{O}(h^{-2})$ for the ultraweak formulation of the Poisson problem.

An interesting observation concerns the dependence of the condition number on the order of approximation. At least for the range of $p = 2, 3, 4, 5$, the results indicate that the condition number of the condensed stiffness matrix is p-independent. A careful comparison of the different formulations, shows that among the DPG formulations the primal formulation gives the worst condition number. On the other hand the ultraweak formulation and the FOSLS give identical condition numbers.

Finally, the standard Galerkin method seems to have an unstable behavior in the pre-asymptotic region but as the solution converges, it recovers the theoretical rate of growth. In fact, in the asymptotic region it delivers the best-conditioned stiffness matrix. In appendix B, a new approach is presented in order to tackle ill-conditioned problems, which delivers $\mathcal{O}(h^{-1})$ growth of the condition number of the DPG system.



(a) $p = 2$ (b) $p = 3$

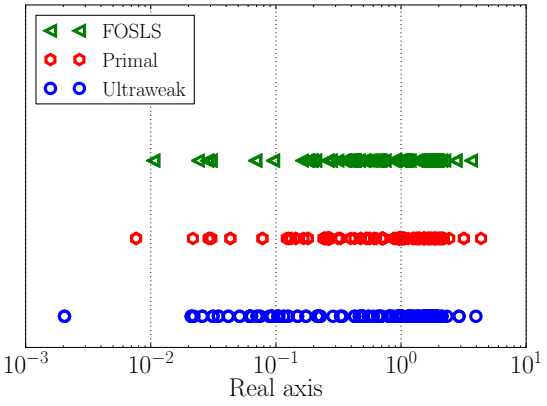


(c) $p = 4$ (d) $p = 5$

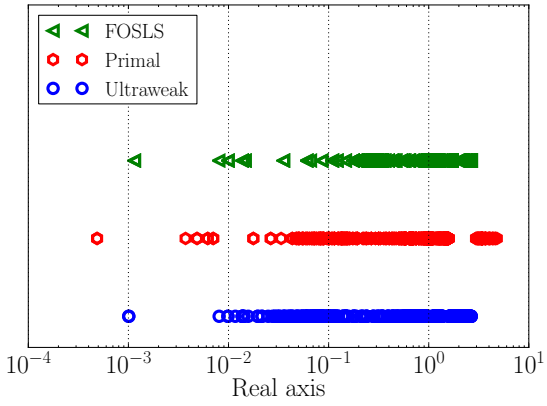
Figure 3.6: Condition number of the DPG matrix resulted from discretization of linear acoustics problem in 2D for various polynomial orders. Here we consider the global stiffness matrix after static condensation of the interior degrees of freedom and diagonal scaling.

3.3.2 Spectrum

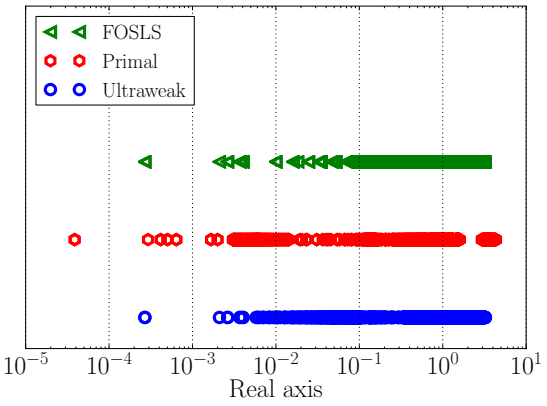
The DPG method is a minimum residual method, and therefore, it always delivers a Hermitian positive definite stiffness matrix. Thus, among many choices of iterative solvers, the Conjugate Gradient (CG) seems to be an ideal candidate for the solution of the DPG system. Theoretical estimates show that convergence of the CG algorithm depends on the condition number, i.e, on the minimum and maximum eigenvalues of the matrix. However, the convergence can be influenced by the whole spectrum. For a detailed analysis we refer the reader to [70].



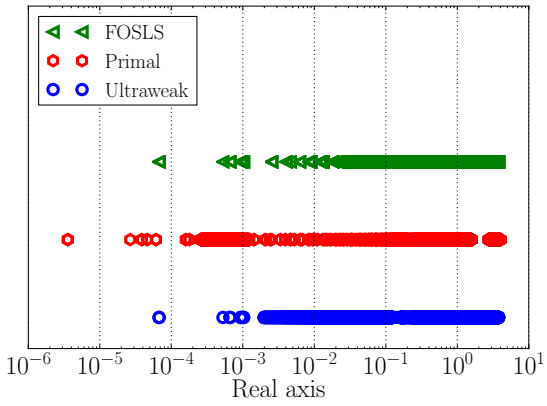
(a) Uniform mesh 4×4 squares



(b) Uniform mesh 8×8 squares



(c) Uniform mesh 16×16 squares



(d) Uniform mesh 32×32 squares

Figure 3.7: Spectrum for the statically condensed DPG system for polynomial order $p = 3$.

We consider the same problem as in Section 3.2.1, where we start with an initial mesh of 16 square elements of polynomial order $p = 3$, and perform three uniform h-refinements. Figure 3.7 shows the evolution of the spectrum of the global stiffness matrix after static condensation of the interior degrees of freedom.

First of all, we can verify that for all the cases the spectrum lies in the positive real axis. This would not be the case for the Galerkin method, when the differential operator is indefinite. In fact, had we used the impedance boundary condition, the spectrum would lie in the complex plane (see Figure 3.8), for a 1D example. Notice that as we refine the mesh the spectrum for the FOSLS and the Ultraweak formulation become very similar. Additionally, if we ignore a few outliers, the spectrum of these two formulations form one big cluster away from zero. A similar behavior can be observed in the one dimensional case (Figure 3.8). This is favorable for the convergence of the CG algorithm (see [70]).

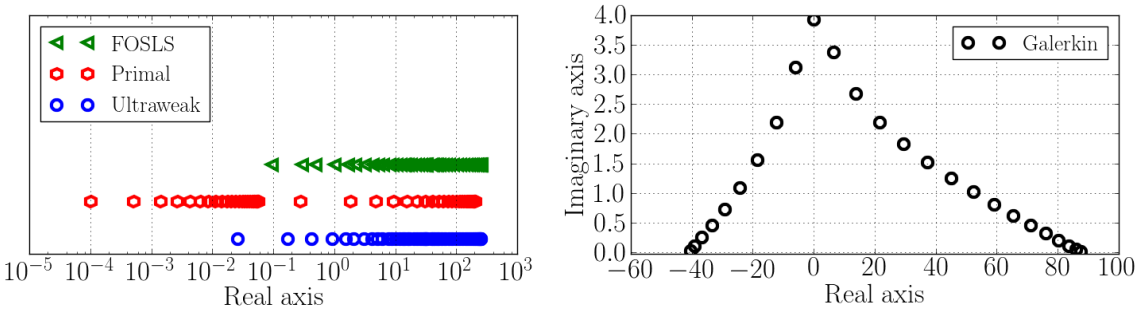


Figure 3.8: Spectrum for 1D linear acoustics with impedance condition. Here, the frequency $\omega = 30$, and the mesh consists of 25 quadratic elements.

3.4 Adaptivity - high frequency beam in two space dimensions

As it was demonstrated in the previous section, the DPG method can be applied to any well posed variational formulation. The ultraweak DPG formulation turns out to be superior compared to the primal DPG formulations and the FOSLS. We note here that the size of the statically condensed system for all DPG formulations and the FOSLS is exactly the same. In

this section we extend our numerical study, by computing with the ultraweak DPG formulation in two space dimensions in the adaptive refinement setting.

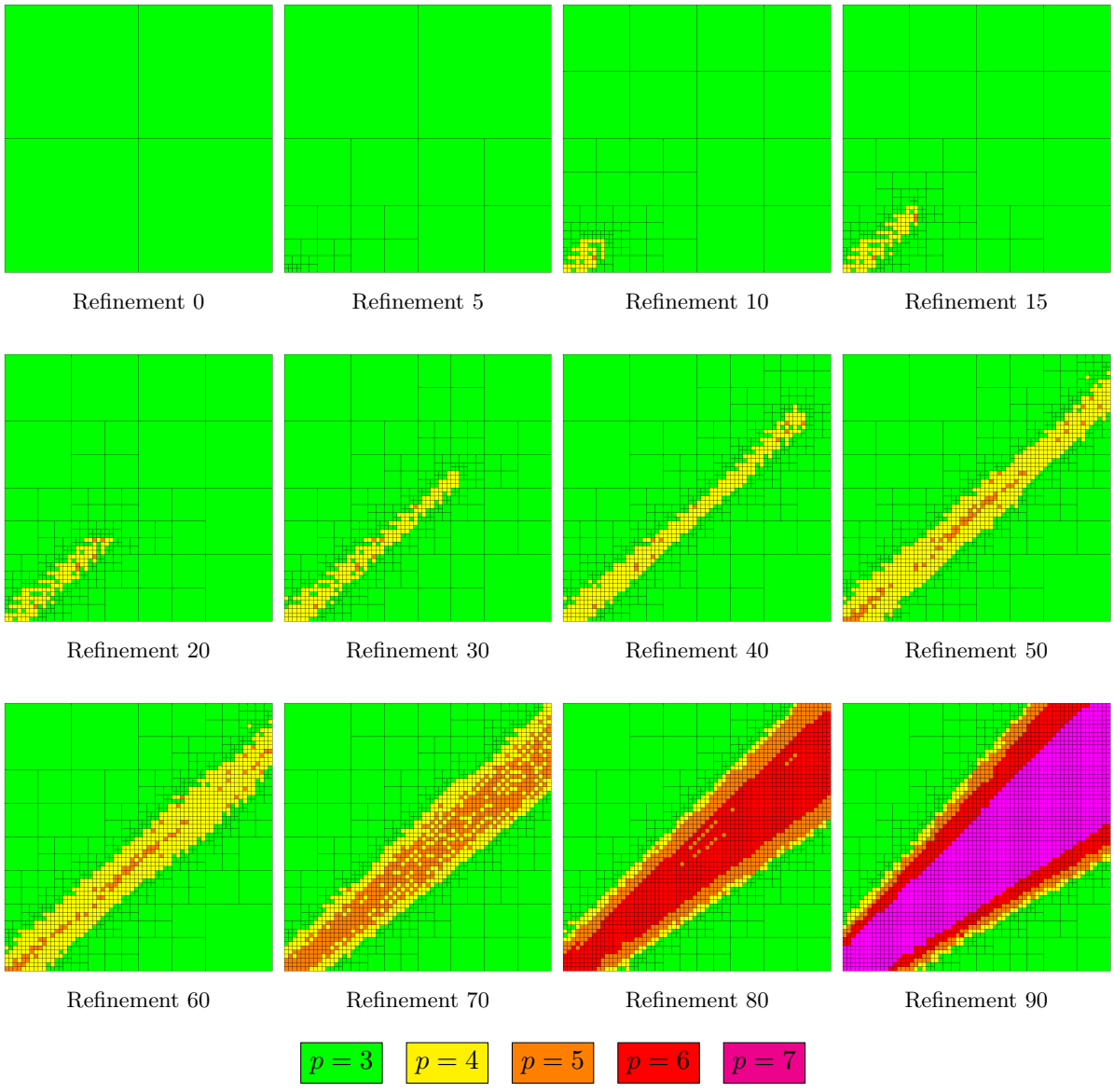


Figure 3.9: Adaptive *hp*-refinements for the simulation of a high-frequency Gaussian beam ($\omega = 120\pi$) in free space using the ultraweak DPG formulation. Observe that the adaptive refinements start from a very coarse mesh and the method produces refinements only in the areas of the domain where there is wave activity.

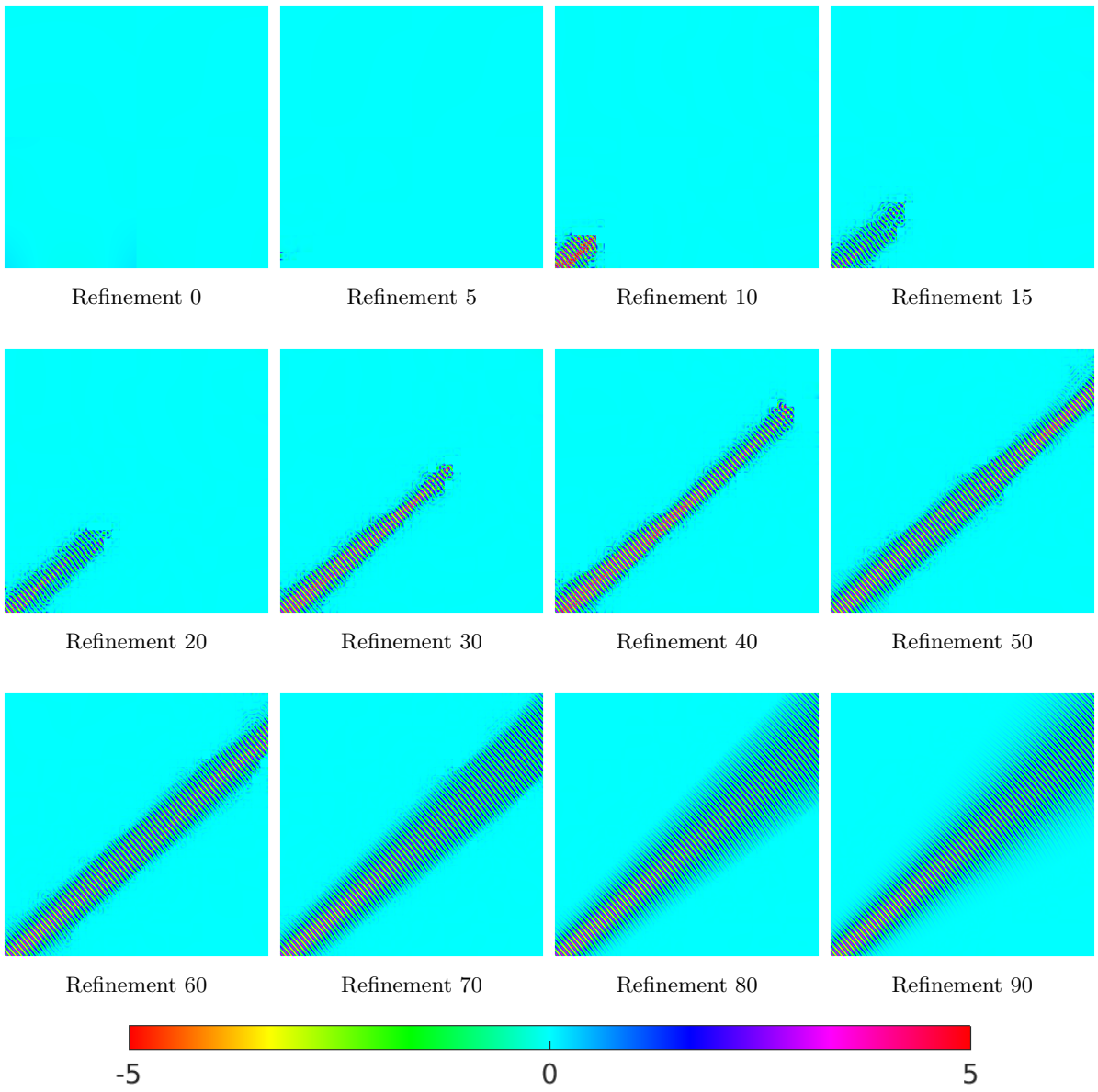


Figure 3.10: Real part of the numerical solution for the acoustic pressure recovered from adaptive mesh refinements. Notice that the solution is built along with the mesh.

We demonstrate the DPG adaptive technology by solving a problem that is characterized by a localized behavior of the solution. We simulate a Gaussian beam propagating in free space. The frequency is 120π (approximately 80 wavelengths along a 40° angle). In

this example the adaptive DPG method turns out to be a very powerful tool, since it avoids unnecessary computations in areas of the domain where the wave is evanescent. In Figures 3.9 and 3.10 we show the evolution of the mesh and the corresponding numerical solution of the real part of the pressure, with the mesh refinements.

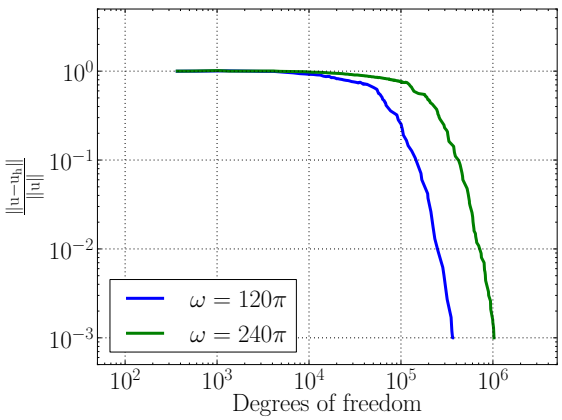
We start the simulation with a mesh of only four square elements of polynomial order of approximation $p = 3$. With this resolution, we are still very far away from satisfying the Nyquist criterion and, as expected, the solution on that mesh is not meaningful. However, even at this stage we can start the adaptive process. The algorithm successfully manages to grow the mesh along with the solution, keeping the mesh very coarse at the areas where the solution is practically zero. We use a simple *hp*-strategy where we perform *h*-refinements until the size of the element reaches half a wavelength and then we switch to *p*-refinements. The polynomial order on an interface edge between two elements is determined by the maximum rule. That is, the order of an edge is set to be the maximum of the order of the neighboring elements. In a scenario in which an edge is constrained (see [25, 37] for implementation of constrained nodes) first the maximum order is passed to the constraining edge. Then the order propagates to the rest of the constrained edges so that constrained and constraining edges have the same order.

3.4.1 Convergence

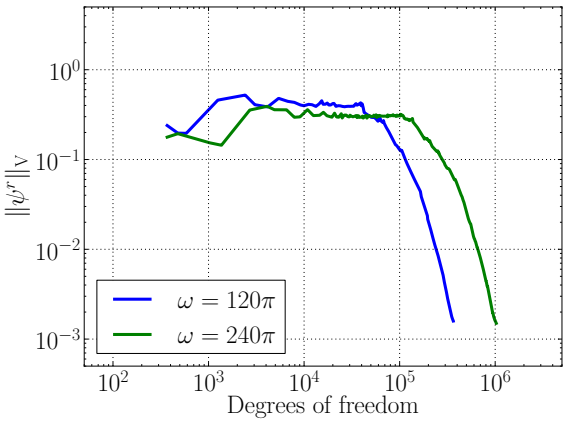
In Figures 3.11a and 3.11b we show the convergence of the global relative error and the global residual respectively, for two different values of the frequency ω. Apart from some variations at the beginning of the simulation, the residual decreases, providing a reliable stopping criterion for the adaptivity process. The relative error decreases monotonically. Additionally, in Figure 3.11c, we plot the ratio of the global residual to the global error. For both values of ω, in the pre-asymptotic region the ratio is approximately of order one. As the approximate solution converges to the exact solution, the ratio tends to one, i.e the global residual gives a very good estimate of the L^2 error.

Finally, in Figure 3.12 we compare the ultraweak DPG method with an adaptive L^2-projection problem. For the solution of the L^2-projection problem, we apply the same *hp*-

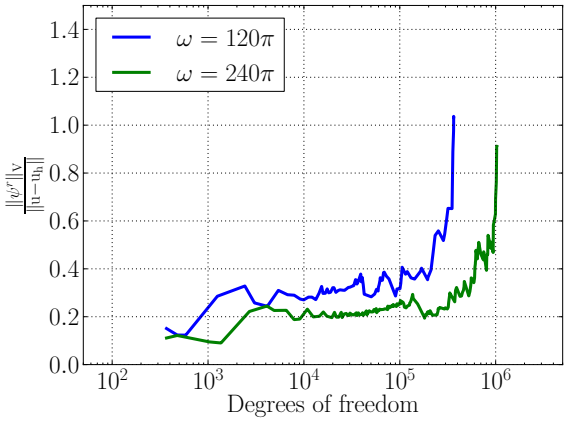
adaptive strategy as for the original problem. Our results show that the ultraweak DPG method, indeed delivers L^2 projection (see Section 3.1.3). Moreover, we can conclude that the adaptive procedure, driven by the residual, is very efficient, i.e., it does produce optimal mesh refinements.



(a) Relative L^2 error

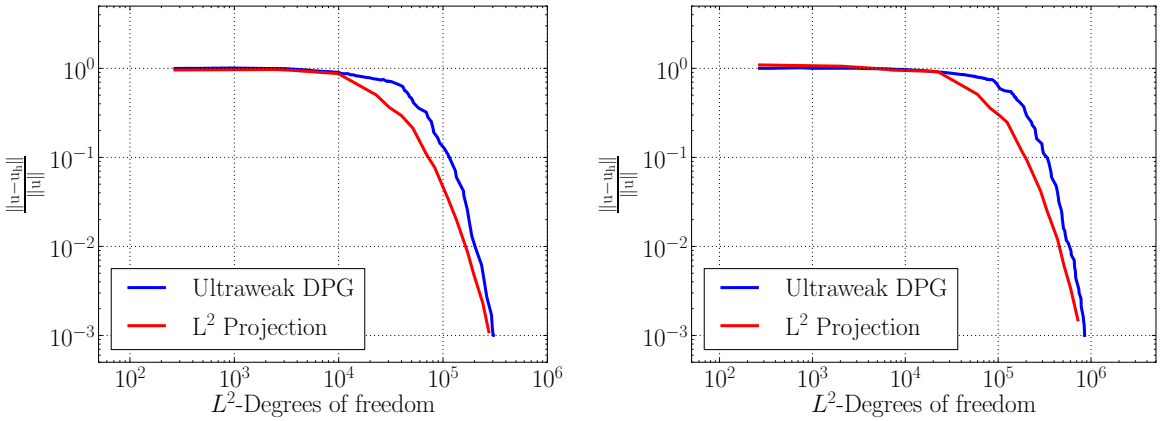


(b) Residual



(c) Ratio

Figure 3.11: Convergence of the adaptive DPG method, for the simulation of high-frequency Gaussian beam in free space using the ultraweak formulation. The figure on the right indicates that the DPG error indicator gives a very good estimate of the actual L^2-error of the method.



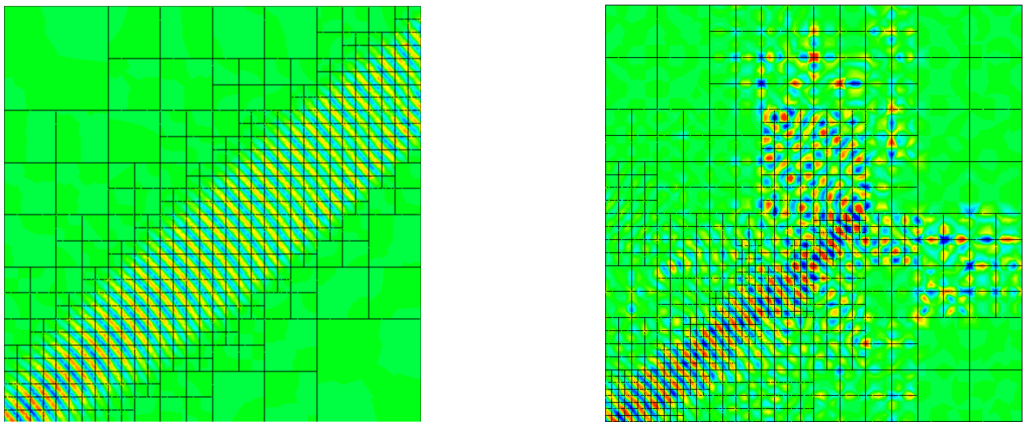
(a) $\omega = 120\pi$ (b) $\omega = 240\pi$

Figure 3.12: Ultraweak DPG error vs L^2-projection error for the simulation of a high-frequency Gaussian beam in free space. This figure shows that the ultraweak DPG formulation delivers L^2-projection.

3.4.2 DPG vs standard FEM

The effect of discrete stability can be clearly observed when we compare the DPG method with the standard Galerkin method. It is well known that the stability of the standard Galerkin method for wave propagation problems can only be proven under the assumption that 'enough' elements per wavelength are used in the discretization. In other words, the method is unstable in the pre-asymptotic region. This makes the standard Galerkin method unsuitable for an adaptive solution scheme that is initiated from very coarse meshes.

In Figure 3.13 we present a qualitative comparison between the ultraweak DPG method and the standard Galerkin method when the same *hp*-adaptive strategy described above is used for the simulation of the Gaussian beam in free space for $\omega = 60\pi$. The error estimator used in the standard Galerkin case is described in detail in [25, 37]. As it can be clearly observed, the standard Galerkin adaptive algorithm fails to capture the wave 'correctly', i.e., additional unnecessary refinements are happening in areas of the domain were there is practically no wave activity. On the other hand, the ultraweak DPG method shows no pre-asymptotic instabilities, and efficiently captures the localized beam. This will turn out to be very important, when designing an adaptive solver for the DPG method.



(a) Ultraweak DPG

(b) Standard FEM

Figure 3.13: Adaptive refinements: ultraweak DPG vs standard FEM. Unlike the standard FEM the DPG method is unconditionally stable, delivering optimal mesh refinements.

At this point we switch gears and focus on the construction of preconditioning techniques for the solution of the DPG linear system. In the following chapter, we present a construction and a theoretical analysis for a one level additive Schwarz preconditioner for the ultraweak DPG formulation when applied to the linear acoustics equations.

Chapter 4

Additive Schwarz preconditioner for the DPG method

We demonstrated in the previous chapter, the ability of the DPG method to efficiently solve problems whose solutions exhibit localized behavior. The DPG adaptive algorithm starts from a very coarse mesh, which could even consist of one element, and uses its built-in local error indicator to develop the mesh along with the solution. However, after every refinement a global system has to be solved. Employing a direct solver at every adaptive step is far from optimal and it is practically unnecessary. After all, at the intermediate meshes, one is interested in a solution "good enough" to provide the next refinement and so a partially converged solution would suffice. It is therefore natural to consider efficient iterative solvers and integrate them within the adaptive process.

Recall that the DPG method delivers a Hermitian (symmetric) and positive definite linear system even if the original problem involves indefinite differential operators. Consequently, a natural choice of an iterative solver is the conjugate gradient (CG) algorithm, provided that we have a good preconditioner. As a stepping stone to a multilevel preconditioner, we start by analyzing a one level domain decomposition preconditioner. This chapter is devoted to the theoretical analysis of such a preconditioner and some numerical results for uniform meshes supporting the analysis.

4.1 Related work on DPG preconditioners

There are several works on preconditioning DPG systems for elliptic problems. To the best of our knowledge the first one chronologically, was done by Barker et al. [7] where the authors analyzed the one level additive Schwarz preconditioner for the Poisson problem. In this work, the authors showcased, both theoretically and numerically, convergence of the

preconditioned CG solver, independent of the mesh size h. A few years after that, Barker et al. in [8] constructed a scalable preconditioner for the primal DPG method again for the Poisson problem. The key point of this implementation was the norm equivalence of the DPG bilinear form with the standard H^1 and $H(\text{div})$ norms. Naturally, existing H^1 and $H(\text{div})$ algebraic multigrid (AMG) technologies could be utilized to precondition the DPG system. A similar approach can also be found in [87], where the authors extended the analysis of [7] for the Poisson problem to the two level setting. Their work was based on well known results on preconditioning the H^1, $H(\text{div})$ and $H(\text{curl})$ spaces [39, 2, 74]. Moreover, a geometric multigrid preconditioner for the Poisson and Stokes problem is presented in [108].

Designing a preconditioner for wave problems is much more challenging. While, it is relatively easy to construct and analyze preconditioners for elliptic problems using the idea of norm equivalence, for wave problems the equivalence constants are frequency dependent. As far as we know there are two works so far attempting to construct robust and efficient preconditioners for wave problems but none of them provide any theoretical results. In [87] the authors present a numerical study on a one level additive Schwarz preconditioner when applied to the Helmholtz problem. Their results indicate uniform convergence of the solver with respect to the frequency ω and the mesh size h. The convergence is however sensitive to the overlap of the Schwarz patch and the number of subdomains. The second attempt on preconditioning the Helmholtz problem is described in [66]. This work introduces a multiplicative Schwarz (block Gauss-Seidel) preconditioner for the primal DPG formulation. The preconditioner is shown to converge at a rate independent of the polynomial order p and the frequency ω.

For our construction we take a slightly different path than most of the works described above. The preconditioner is constructed directly by the underlying bilinear form of the wave problem. Using the properties of the ultraweak DPG method we invoke a numerical experiment to examine the dependence of the condition number of the preconditioned system with respect to the polynomial order of approximation, the frequency, the Schwarz patch size and the mesh size. Even though the analysis is done only for a one level additive preconditioner, it gives us useful insights for the construction of a multilevel preconditioner. We note that our results are

consistent with the results of related work described above.

4.2 Preliminaries

As a prelude to the construction of our preconditioner for the ultraweak DPG formulation, we present some well known fundamental results used in the analysis of preconditioning [1]. In this section the notation is independent of the rest of the document.

4.2.1 Notation and fundamental results

Let U be a Hilbert space, and let $b(u,v)$ and $a(u,v)$ be two self adjoint positive definite forms and $l(v)$ a linear form defined on U. Let $\{e_i\}_{i=1}^N$ be a basis for a finite-dimensional subspace $U_h \subset U$. We denote by $\mathsf{A}_{ij}, \mathsf{B}_{ij}$ the Galerkin stiffness matrices corresponding to forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ and by l_j the load vector corresponding to the linear form $l(\cdot)$, i.e.,

$$\mathsf{A}_{ij} = a(e_j, e_i), \quad \mathsf{B}_{ij} = b(e_j, e_i), \quad \mathsf{l}_j = l(e_j), \quad i,j = 1,\dots,N.$$

We would like to solve the system

$$\mathsf{B}\mathsf{u} = \mathsf{l} \tag{4.1}$$

and the intent is to precondition it with A, i.e, precondition $b(u,v)$ with $a(u,v)$. Note that since A and B are self adjoint and positive definite, they define inner products on U_h, i.e.,

$$(\cdot,\cdot)_{\mathsf{A}} = (\mathsf{A}\cdot,\cdot),$$
$$(\cdot,\cdot)_{\mathsf{B}} = (\mathsf{B}\cdot,\cdot)$$

The induced norms are given by $\|\cdot\|_{\mathsf{A}}$ and $\|\cdot\|_{\mathsf{B}}$. Consider now the following linear iteration scheme to solve (4.1).

$$\mathsf{u}_{k+1} = \mathsf{u}_k + \mathsf{A}(\mathsf{l} - \mathsf{B}\mathsf{u}_k), \quad k = 0, 1\dots \tag{4.2}$$

[1] The author would like to express his gratitude to Jay Gopalakrishnan for providing helpful reading material [61] and for the invaluable discussions on the subject of preconditioning.

where u_0 is a given initial guess. Note that for $\mathsf{A} = \mathsf{B}^{-1}$, the iteration scheme converges in one iteration. Let u be the solution to the linear system $\mathsf{Bu} = \mathsf{l}$. Then

$$\underbrace{\mathsf{u} - \mathsf{u}_k}_{\mathsf{e}_k} = \underbrace{(\mathsf{I} - \mathsf{AB})}_{\mathsf{E}}^k \underbrace{(\mathsf{u} - \mathsf{u}_0)}_{\mathsf{e}_0}.$$

We call the operator $\mathsf{E} = \mathsf{I} - \mathsf{AB}$ the *reducer* and the operator A the *iterator*. The iteration scheme, can be used directly as a solver and if the norm of the reducer is less than one, it would be convergent. The following proposition gives sufficient conditions for convergence.

Proposition 4.1. *The linear iteration*

$$\mathsf{u}_{k+1} = \mathsf{u}_k + \theta \mathsf{A}(\mathsf{l} - \mathsf{B}\mathsf{u}_k) \tag{4.3}$$

is convergent for $0 < \theta < 2/\lambda_{\max(\mathsf{AB})}$ *with an optimal error reduction per iteration of*

$$\frac{\kappa(\mathsf{AB}) - 1}{\kappa(\mathsf{AB}) + 1} \text{ when } \theta = \frac{2}{\lambda_{\min}(\mathsf{AB}) + \lambda_{\max}(\mathsf{AB})}$$

The proof of the above result can be found in [110].

The iteration scheme (4.3) can be used as a preconditioner for the Conjugate Gradient algorithm. The following two theorems show that in such a case the convergence is accelerated. First, the general convergence estimate of the CG solver is given by the following theorem.

Theorem 4.1. *(Conjugate gradient (CG) convergence)*
Let B *be Hermitian (symmetric) positive definite with respect to a certain inner product. Then the error in the* k^{th} *iteration of the CG algorithm measured in the* $\mathsf{B}-$*norm is given by*

$$\|u - u_k\|_{\mathsf{B}} \leq 2\left(\frac{\sqrt{\kappa(\mathsf{B})} - 1}{\sqrt{\kappa(\mathsf{B})} + 1}\right)^k \|u - u_0\|_{\mathsf{B}}$$

Here $\kappa(\mathsf{B}) = \dfrac{\lambda_{\max}(\mathsf{B})}{\lambda_{\min}(\mathsf{B})}$ *is the condition number of* B*. Note that the min and max eigenvalues are given by:*

$$\lambda_{\max}(\mathsf{B}) = \max_{\substack{v \in U_h \\ v \neq 0}} \frac{(\mathsf{B}v, v)}{\|v\|^2}, \qquad \lambda_{\min}(\mathsf{B}) = \min_{\substack{v \in U_h \\ v \neq 0}} \frac{(\mathsf{B}v, v)}{\|v\|^2}$$

A proof of the above result can be found in many books (see [110, 70]). Finally, the following theorem shows, how the convergence of the linear iteration (4.3) can be accelerated by the CG solver. A proof of the theorem is given in Appendix C.1

Theorem 4.2. *(Iterator as a preconditioner)*
Consider the linear iteration (4.2) *and suppose* A *and* B *are self-adjoint with respect to* $(\cdot,\cdot)$. *If* B *is bounded, bijective, positive definite and*

$$\eta = \|\mathrm{I} - \mathsf{AB}\|_{\mathsf{B}} < 1$$

then

1. A *is positive definite.*
2. *Iteration* (4.2) *is convergent.*
3. *The condition number* $\kappa(\mathsf{AB}) = \dfrac{\lambda_{\max}(\mathsf{AB})}{\lambda_{\min}(\mathsf{AB})}$ *satisfies* $\kappa(\mathsf{AB}) \leq \dfrac{1+\eta}{1-\eta}$.
4. *The asymptotic convergence rate of the conjugate gradient method for the preconditioned system is faster than the rate of convergence of* (4.2).

Since, the CG algorithm is invariant with respect to the relaxation parameter θ, we can always assume that the optimal θ is used and therefore the linear iteration (4.3) is a contraction (the norm of the reducer is less than one). In view of the result of Theorem 4.2, we conclude that CG algorithm accelerates the convergence of the linear iteration (4.3). Nevertheless, the convergence of the CG algorithm is directly related to the condition number of the preconditioned system. Therefore if we are interested in preconditioning the matrix B with matrix A, we need an estimate for the condition number $\kappa(\mathsf{AB})$.

The general philosophy behind preconditioning is to prove an equivalence relation between specific norms, involving both the original bilinear form of the problem and the bilinear form induced by the preconditioner. Note that the preconditioner's bilinear form has to remain self-adjoint and coercive. We outline next, two well established theories, that provide tools for proving the norm equivalence. The first one, Nepomnyaschikh's fictitious space lemma [99], is very general, since the result relates any two self-adjoint and coercive bilinear forms. The second one, *subspace correction theory*, was developed by Jinchao Xu [117]. It provides the mathematical foundation for convergence analysis of solvers that involve subspace solvers such as domain decomposition and multigrid. We note that for the one level Schwarz preconditioner the Nepomnyaschikh's lemma can be easily adapted to the subspace correction theory.

4.2.2 Norm equivalence and Nepomnyaschikh's theorem

Lemma 4.1. *(Norm equivalence)*

Assume that A *and* B *are both self-adjoint and positive definite with respect to* $(\cdot,\cdot)$ *and* c_0, c_1 *are positive constants. Then the following are equivalent* $\forall\, v \in U_h$*:*

$$c_0(\mathsf{A}v, v) \leq (\mathsf{ABA}v, v) \leq c_1(\mathsf{A}v, v), \tag{4.4a}$$

$$c_0(\mathsf{B}v, v) \leq (\mathsf{BAB}v, v) \leq c_1(\mathsf{B}v, v), \tag{4.4b}$$

$$c_1^{-1}(\mathsf{A}v, v) \leq (B^{-1}v, v) \leq c_0^{-1}(\mathsf{A}v, v), \tag{4.4c}$$

$$c_1^{-1}(\mathsf{B}v, v) \leq (A^{-1}v, v) \leq c_0^{-1}(\mathsf{B}v, v). \tag{4.4d}$$

Additionally, the condition number $\kappa(\mathsf{AB}) \leq \dfrac{c_1}{c_0}$.

Lemma 4.2. *(Nepomnyaschikh fictitious space lemma [99]) Let* X, Y*, be Hilbert spaces with inner products* $(\cdot,\cdot)_X$ *and* $(\cdot,\cdot)_Y$ *respectively. Define the following two sesquilinear, continuous, Hermitian and coercive forms defined on* X *and* Y *respectively.*

$$b(x, \delta x) = \langle \mathrm{B}x, \delta x\rangle_{X'\times X} \qquad \textit{with } \mathrm{B} : X \to X'$$

$$a_0(y, \delta y) = \langle \mathrm{A}_0 y, \delta y\rangle_{Y'\times Y} \qquad \textit{with } \mathrm{A}_0 : Y \to Y'$$

Additionally we assume the existence of a continuous surjective operator $\mathrm{R} : Y \to X$ *and a continuous injective operator* $\mathrm{T} : X \to Y$ *that satisfy*

$$\mathrm{R} \circ \mathrm{T} = \mathrm{id}_X \quad \textit{i.e.} \quad \mathrm{RT}x = x, \ \ \forall x \in X.$$

Take now an arbitrary $l \in X'$*, and consider the following two variational problems:*

$$\begin{cases} \textit{Find } x \in X : \\ b(x, \delta x) = \langle l, \delta x\rangle \quad \forall\, \delta x \in X \end{cases} \quad \textit{and} \quad \begin{cases} \textit{Find } y \in Y : \\ a_0(y, \delta y) = \langle \mathrm{R}' l, \delta y\rangle = \langle l, \mathrm{R}\delta y\rangle \quad \forall\, \delta y \in Y \end{cases} \tag{4.5}$$

and assume the following two inequalities:

$$\exists\, c_{\mathrm{R}} > 0 : \forall y \in Y \qquad b(\mathrm{R}y, \mathrm{R}y) \leq c_{\mathrm{R}}^2 a_0(y, y) \qquad \textit{and}$$

$$\exists\, d_{\mathrm{R}} > 0 : \forall x \in X \quad a_0(\mathrm{T}x, \mathrm{T}x) \leq d_{\mathrm{R}}^{-2} b(x, x)$$

Now let $l \in Y'$ and x, y be the solutions of the variational problems in (4.5). *Then*

$$c_{\mathrm{R}}^{-2} a_0(y,y) \le b(x,x) \le d_{\mathrm{R}}^{-2} a_0(y,y) \tag{4.7}$$

Note that in the finite dimensional case for $x = \mathrm{B}^{-1}l$, $y = \mathrm{A}_0^{-1}\mathrm{R}'l$ and $\mathrm{A} := \mathrm{R}\mathrm{A}_0^{-1}\mathrm{R}'$, (4.7) *is equivalent with* (4.4b) *and that gives an upper bound for the condition number $\kappa(\mathrm{AB}) \le \dfrac{d_{\mathrm{R}}^2}{c_{\mathrm{R}}^2}$.*

We refer the reader to Sections C.1.1 and C.1.2 for the proofs of the above two lemmas.

4.2.3 Additive Schwarz preconditioner and the subspace correction theory

Let B be self-adjoint and positive definite defined on the vector space U. Suppose that U_i, $i = 1, \dots J$, are closed subspaces of the Hilbert space $(U, (\cdot,\cdot))$. Additionally, let B_i be self-adjoint and positive definite operator defined on U_i by

$$(\mathsf{B}_i \mathsf{u}_i, \mathsf{v}_i) = (\mathsf{B}\mathsf{u}_i, \mathsf{v}_i), \quad \forall \mathsf{u}_i, \mathsf{v}_i \in U_i,$$

and let $\mathsf{Q}_i : U \to U_i$ denote the $(\cdot,\cdot)$-orthogonal projection onto U_i. Then, the operator

$$\mathsf{A} = \sum_{i=1}^{J} \mathsf{B}_i^{-1} \mathsf{Q}_i$$

is called the *additive* preconditioner based on subspaces U_i and operators B_i. The additive Schwarz algorithm is given by:

Algorithm 1 Additive Schwarz/parallel subspace correction

1: **procedure** PSC($\mathsf{u}_k, \mathsf{u}_{k+1}$) ▷ Given u_k compute u_{k+1}
2: $\mathsf{r} = \mathsf{l} - \mathsf{B}\mathsf{u}_k$ ▷ Compute initial residual
3: $\mathsf{r}_i = \mathsf{Q}_i \mathsf{r}$ ▷ Project the residual on to U_i
4: $\mathsf{B}_i \mathsf{z}_i = \mathsf{r}_i$ ▷ Solve on the subspace the local problem
5: $\mathsf{u}_{k+1} = \mathsf{u}_k + \theta \sum_{i=i}^{J} \mathsf{z}_i$ ▷ Correct u_k on each subspace

Theorem 4.3 (*Subspace correction*)**.** *For the above setting, assume the following two statements hold:*

- *(strengthened Cauchy–Schwarz inequality) there exists a number* $\beta > 0$ *such that for all* $\mathsf{u}_i, \mathsf{v}_i \in U_i$

$$\sum_{i=1}^{J}\sum_{j=1}^{J} |(\mathsf{u}_i, \mathsf{v}_j)_{\mathsf{B}}| \leq \beta \Big(\sum_{i=1}^{J} \|\mathsf{u}_i\|_{\mathsf{B}}^2 \Big)^{1/2} \Big(\sum_{j=1}^{J} \|\mathsf{v}_j\|_{\mathsf{B}}^2 \Big)^{1/2}$$

- *(stable decomposition) there exists a number* $\alpha > 0$ *such that* $\forall \mathsf{u} \in U$*, there exists a decomposition* $\mathsf{u} = \sum_{i=1}^{J} \mathsf{u}_i$*, with* $\mathsf{u}_i \in U_i$*, that satisfies*

$$\sum_{i=1}^{J} \|\mathsf{u}_i\|_{\mathsf{B}}^2 \leq \alpha^{-1} \|\mathsf{u}\|_{\mathsf{B}}^2.$$

Then, the following equivalence relation is true:

$$\alpha(\mathsf{u}, \mathsf{u})_{\mathsf{B}} \leq (\mathsf{P}\mathsf{u}, \mathsf{u})_{\mathsf{B}} \leq \beta(\mathsf{u}, \mathsf{u})_{\mathsf{B}} \tag{4.8}$$

where $\mathsf{P} = \sum_{i=1}^{J} \mathsf{P}_i$ *and* $\mathsf{P}_i : U \to U_i$ *is the* $(\cdot,\cdot)_{\mathsf{B}}$*-orthogonal projector, i.e.,*

$$(\mathsf{P}_i \mathsf{u}, \mathsf{v}_i)_{\mathsf{B}} = (\mathsf{u}, \mathsf{v}_i)_{\mathsf{B}}, \quad \forall \mathsf{v}_i \in U_i.$$

Note that $\mathsf{B}_i \mathsf{P}_i = \mathsf{Q}_i \mathsf{B}$ and therefore $\mathsf{P} = \mathsf{A}\mathsf{B}$ is the preconditioned matrix. Consequently, (4.8) along with Lemma 4.1 give an estimate of the condition number $\kappa(\mathsf{A}\mathsf{B}) = \beta/\alpha$. A detailed proof of this theorem can be found in [117]. Note that Lemma 4.2 can be reduced to Theorem 4.3 in the subspace correction setting. In particular, it is easy to see that the operator T in (4.6) is exactly the stable decomposition assumption of Theorem 4.3.

4.2.4 A Schur complement result

The following result, found also in [99], is useful when proving energy estimates for Schur complements. This result is relevant in our work, because in practice the interior degrees of freedom are condensed out of the final system. In the case of the ultraweak formulation, static condensation is essential, since all the L^2 variables can be eliminated in an element-wise fashion. Consequently, the final linear system has a significantly reduced size (for high order approximations the size could be reduced by $70 - 80\%$) and involves only the interface unknowns. In summary, this result guarantees that the bound of the condition number of the condensed system is the same with the bound of the condition number of the original system.

This allows, for the analysis to be done on the original system, but the implementation on the condensed system. The result is summarized by the two lemmas below, and their proofs are given in Appendix C.2.

Let U be a Hilbert space given as a Cartesian product of two Hilbert spaces U_1, U_2. Let $u = (u_1, u_2), v = (v_1, v_2)$. Assume, we are given a sesquilinear, continuous, Hermitian form:

$$b(u, v) = b_{11}(u_1, v_1) + b_{12}(u_2, v_1) + b_{21}(u_1, v_2) + b_{22}(u_2, v_2)\,.$$

Let $\mathrm{B}, \mathrm{B}_{11}, \mathrm{B}_{12}, \mathrm{B}_{21}, \mathrm{B}_{22}$ be the corresponding operators,

$$\begin{aligned}
&\mathrm{B} : U \to U' && \langle \mathrm{B}u, v\rangle && = b(u, v) && u \in U,\ v \in U\\
&\mathrm{B}_{11} : U_1 \to U_1' && \langle \mathrm{B}_{11}u_1, v_1\rangle && = b_{11}(u_1, v_1) && u_1 \in U_1,\ v_1 \in U_1\\
&\mathrm{B}_{12} : U_2 \to U_1' && \langle \mathrm{B}_{12}u_2, v_1\rangle && = b_{12}(u_2, v_1) && u_2 \in U_2,\ v_1 \in U_1\\
&\mathrm{B}_{21} : U_1 \to U_2' && \langle \mathrm{B}_{21}u_1, v_2\rangle && = b_{21}(u_1, v_2) && u_1 \in U_1,\ v_2 \in U_2\\
&\mathrm{B}_{22} : U_2 \to U_2' && \langle \mathrm{B}_{22}u_2, v_2\rangle && = b_{22}(u_2, v_2) && u_2 \in U_2,\ v_2 \in U_2.
\end{aligned}$$

Lemma 4.3. *Assume additionally that b_{11} is positive definite. This implies that B_{11} is invertible and the following identity holds:*

$$\inf_{u_1 \in U_1} b((u_1, u_2), (u_1, u_2)) = \inf_{u_1 \in U_1} \langle \mathrm{B}(u_1, u_2), (u_1, u_2)\rangle = \langle (\mathrm{B}_{22} - \mathrm{B}_{21}\mathrm{B}_{11}^{-1}\mathrm{B}_{12})u_2, u_2\rangle\,.$$

Let $a(u, v)$ be now another sesquilinear, continuous, Hermitian, semi-positive form on U with a positive-definite part a_{11} as well. Denote the two Schur complement operators by:

$$\mathrm{S}_\mathrm{A} := \mathrm{A}_{22} - \mathrm{A}_{21}\mathrm{A}_{11}^{-1}\mathrm{A}_{12}, \qquad \mathrm{S}_\mathrm{B} := \mathrm{B}_{22} - \mathrm{B}_{21}\mathrm{B}_{11}^{-1}\mathrm{B}_{12}\,.$$

Lemma 4.4. *Assume forms a and b (or operators A and B) are positive semi-definite and spectrally equivalent, i.e.,*

$$c_1\, a(u, u) \le b(u, u) \le c_2\, a(u, u) \quad \Leftrightarrow \quad c_1\, \langle \mathrm{A}u, u\rangle \le \langle \mathrm{B}u, u\rangle \le c_2\, \langle \mathrm{A}u, u\rangle\,,$$

with some positive constants c_1, c_2. Then the corresponding Schur complement operators are spectrally equivalent on U_2 with the same constants,

$$c_1\, \langle \mathrm{S}_\mathrm{A}u_2, u_2\rangle \le \langle \mathrm{S}_\mathrm{B}u_2, u_2\rangle \le c_2\, \langle \mathrm{S}_\mathrm{A}u_2, u_2\rangle\,.$$

4.3 Analysis of the preconditioner - one level setting

It should be clear by now that in order to estimate the condition number of the preconditioned system the only thing we need to do, is to verify the two assumptions of Theorem 4.3. While verifying the first one is standard, the second one (stable decomposition) is not so trivial. We present our analysis in the forthcoming section. First, we give a small recap of the DPG ultraweak formulation.

4.3.1 Preconditioning the ultraweak formulation

Consider an arbitrary first-order system of PDEs defined on a bounded and simply connected domain Ω, expressed in the abstract form:

$$\begin{cases} \mathfrak{u} \in D(\mathrm{A}) \\ \mathrm{A}\mathfrak{u} = \mathfrak{f} \end{cases} \tag{4.9}$$

where A is some differential operator and $D(\mathrm{A})$ denotes it's domain. For instance, recall from (3.1), the strong formulation of the time-harmonic linear acoustics equations with impedance boundary condition:

$$\begin{aligned} &\mathfrak{u} = (p, u) \\ &\mathrm{A}((p,u)) = (i\omega p + \operatorname{div} u, i\omega u + \nabla p) \\ &\mathfrak{f} = (f_1, f_2) \in L^2(\Omega) \times (L^2(\Omega))^d =: \boldsymbol{L^2}(\Omega) \end{aligned}$$

and the domain of A is given by

$$D(\mathrm{A}) := \{ \mathfrak{u} \in \boldsymbol{L^2}(\Omega) \ : \mathrm{A}\mathfrak{u} \in \boldsymbol{L^2}(\Omega),\, p - u \cdot n \ = 0 \text{ on } \partial\Omega \ \}$$

The ultraweak variational formulation of problem (4.9) is:

$$\begin{cases} \mathfrak{u} \in \boldsymbol{L^2}(\Omega) \\ (\mathfrak{u}, \mathrm{A}^*\mathfrak{v}) = (\mathfrak{f}, \mathfrak{v}) \quad \mathfrak{v} \in D(\mathrm{A}^*). \end{cases}$$

where $D(\mathrm{A}^*)$ is equipped with the adjoint graph norm,

$$\|\mathfrak{v}\|_V^2 := \|\mathrm{A}^*\mathfrak{v}\|^2 + \alpha^2 \|\mathfrak{v}\|^2 \,,$$

with a scaling parameter $\alpha = \mathcal{O}(1)$. Here $\mathfrak{v}$ denotes a group test variable, i.e, $\mathfrak{v} := (q, v)$. For the linear acoustics problem, the operator is (formally) *skew-adjoint*, $\mathrm{A}^* = -\mathrm{A}$, and

$$D(\mathrm{A}^*) = \{\mathfrak{v} \in \boldsymbol{L}^2(\Omega) \,:\, \mathrm{A}^*\mathfrak{v} \in \boldsymbol{L}^2(\Omega) : q + v \cdot n \;= 0 \text{ on } \partial\Omega\}$$

The boundary conditions on the test functions (built into the definition of $D(\mathrm{A}^*)$), can be eliminated at the expense of introducing extra unknowns $\hat{\mathfrak{u}} := (\hat{p}, \hat{u}_n)$. We arrive then at the linear problem

$$\begin{cases} \mathfrak{u} \in \boldsymbol{L}^2(\Omega),\, \hat{\mathfrak{u}} \in \hat{U} \\ (\mathfrak{u}, \mathrm{A}^*\mathfrak{v}) + \langle \hat{\mathfrak{u}}, \mathfrak{v} \rangle = (\mathfrak{f}, \mathfrak{v}), \quad \mathfrak{v} \in H_{\mathrm{A}^*}(\Omega)\,, \end{cases} \tag{4.10}$$

where

$$H_{\mathrm{A}^*}(\Omega) := \{\mathfrak{v} \in \boldsymbol{L}^2(\Omega) \,:\, \mathrm{A}^*\mathfrak{v} \in \boldsymbol{L}^2(\Omega)\}$$

is equipped with the same graph norm as above. The flux $\hat{\mathfrak{u}}$ is a trace of a function that lives in the original energy graph space,

$$\hat{U} := \operatorname{tr} D(\mathrm{A})\,.$$

In particular, for the acoustics problem, the trace space is defined by:

$$\begin{aligned} \hat{U} &= \{(\hat{p}, \hat{u}_n) \,:\, \text{exists } (p, u) \in D(\mathrm{A}) \text{ such that } (\hat{p}, \hat{u}_n) = \operatorname{tr}(p, u)\} \\ &= \{(\hat{p}, \hat{u}_n) \in H^{\frac{1}{2}}(\partial\Omega) \times H^{-\frac{1}{2}}(\partial\Omega) \,:\, \hat{p} - \hat{u}_n = 0 \text{ on } \partial\Omega\}\,. \end{aligned}$$

and it is equipped with the minimum energy extension norm,

$$\|\hat{\mathfrak{u}}\|_{\hat{U}} := \inf_{\substack{\mathfrak{u} \in D(\mathrm{A}) \\ \operatorname{tr}\mathfrak{u} = \hat{\mathfrak{u}}}} \|\mathfrak{u}\|_{H_{\mathrm{A}}(\Omega)}$$

where

$$\|\mathfrak{u}\|^2_{H_{\mathrm{A}}(\Omega)} := \|\mathrm{A}\mathfrak{u}\|^2 + \|\mathfrak{u}\|^2\,.$$

The corresponding ultraweak variational formulation with broken test space has a similar structure as formulation (4.10),

$$\begin{cases} \mathfrak{u} \in \boldsymbol{L}^2(\Omega),\, \hat{\mathfrak{u}} \in \hat{U}_h \\ \underbrace{(\mathfrak{u}, \mathrm{A}^*\mathfrak{v}) + \langle \hat{\mathfrak{u}}, \mathfrak{v} \rangle}_{=: b((\mathfrak{u}, \hat{\mathfrak{u}}), \mathfrak{v})} = (\mathfrak{f}, \mathfrak{v}) \quad \mathfrak{v} \in V(\Omega_h)\,. \end{cases}$$

The broken space equals

$$V(\Omega_h) := \{v \in \boldsymbol{L}^2(\Omega) \,:\, \mathrm{A}_h^* \mathfrak{v} \in \boldsymbol{L}^2(\Omega)\}$$

and it is equipped again with the adjoint graph norm,

$$\|\mathfrak{v}\|^2_{V(\Omega_h)} := \sum_K \underbrace{(\|\mathrm{A}_h^* \mathfrak{v}\|^2 + \|\mathfrak{v}\|^2)}_{=:\|\mathfrak{v}\|^2_{V(K)}} .$$

where the h in symbol A_h^* indicates that the operator is understood *element-wise*. Fluxes and traces are defined now on the whole mesh skeleton Γ_h and equipped again with the minimum energy extension norm. The trace space is given by

$$\hat{U}_h = \{(\hat{p}, \hat{u}_n) \in H^{\frac{1}{2}}(\Gamma_h) \times H^{-\frac{1}{2}}(\Gamma_h) \,:\, \hat{p} - \hat{u}_n = 0 \text{ on } \partial\Omega\} .$$

Recall that the *energy norm* is given by

$$\|(\mathfrak{u}, \hat{\mathfrak{u}})\|^2_E := \left(\sup_{\mathfrak{v} \in V(\Omega_h)} \frac{b((\mathfrak{u}, \hat{\mathfrak{u}}), \mathfrak{v})}{\|\mathfrak{v}\|_{V(\Omega_h)}} \right)^2 = \sum_K \underbrace{\left(\sup_{\mathfrak{v} \in V(K)} \frac{|(\mathfrak{u}, \mathrm{A}_h^* \mathfrak{v}) + \langle \hat{\mathfrak{u}}, \mathfrak{v} \rangle|}{\|\mathfrak{v}\|_{V(K)}} \right)^2}_{=:\|(\mathfrak{u}, \hat{\mathfrak{u}})\|^2_{E,K}} \tag{4.11}$$

and enjoys the same property as standard Sobolev energy norms - the global norm (squared) equals the sum of element norms (squared) [19]. This result allows us to consider a single element, when proving norm equivalence. We are now ready to present the analysis.

4.3.2 Set up

Let $\mathrm{u} = (\mathfrak{u}, \hat{\mathfrak{u}})$ denote the group variable including the field and trace variables of the ultraweak formulation. We want to precondition the energy norm $\|\cdot\|_E$ given by (4.11). Let $b_E(\cdot, \cdot)$ be the Hermitian and coercive form corresponding to the energy norm, i.e.,

$$b_E(\mathrm{u}, \mathrm{v}) = b(\mathrm{u}, \mathrm{Tv}) = (\mathrm{Tu}, \mathrm{Tv})_V$$

where T is the trial-to-test operator defined in (2.11). We introduce $\{\Omega_i\}_{i=1}^J$ a finite cover of Ω such that each $\bar{\Omega}_i$ is the support of a vertex shape function. We denote the size of a vertex patch by δ. In addition, we assume that the cover has the *finite overlap property*, i.e.,

there exists an integer r such that each point of Ω is contained in at most r of the sets Ω_i. Equivalently, let χ_i be the characteristic function of Ω_i. Then

$$\sum_{i=1}^{J} \|\chi_i\|_{L^\infty}^2 = \sum_{i=1}^{J} \|\chi_i\|_{L^\infty} \leq r \tag{4.12}$$

Finally, the local energy subspaces corresponding to the partition are given by:

$$U_i = \boldsymbol{L}^2(\Omega_i) \times \hat{U}_i,$$

where $\hat{U}_i := \{\hat{\mathfrak{u}} \in \hat{U}_h : \hat{\mathfrak{u}} = 0 \text{ on } \Gamma_h - \Omega_i\}$, and Γ_h is the mesh skeleton.

4.3.3 Strengthened Cauchy–Schwarz inequality

Verifying the first assumption, of Theorem 4.3 is straight forward. Let $\mathsf{u}_i \in U_i$ and $\mathsf{v}_j \in U_j$. Then, for any inner product $(\cdot,\cdot)_B$ the inequality

$$|(\mathsf{u}_i, \mathsf{v}_j)_{\mathsf{B}}| \leq \varepsilon_{ij} \|\mathsf{u}_i\|_{\mathsf{B}} \|\mathsf{v}_j\|_{\mathsf{B}}$$

is true for $\varepsilon_{ij} \leq 1$. Indeed, from Cauchy–Schwarz inequality, $\varepsilon_{ij} = 1$ when $(\mathsf{u}_i, \mathsf{v}_j)_{\mathsf{B}} \neq 0$, but it can be chosen to be 0 when $(\mathsf{u}_i, \mathsf{v}_j)_{\mathsf{B}} = 0$. Taking the sum over i and j we have:

$$\begin{aligned}\sum_{i=1}^{J}\sum_{j=1}^{J} |(\mathsf{u}_i, \mathsf{v}_j)_{\mathsf{B}}| &\leq \sum_{i=1}^{J}\sum_{j=1}^{J} \varepsilon_{ij} \|\mathsf{u}_i\|_{\mathsf{B}} \|\mathsf{v}_j\|_{\mathsf{B}} \\ &\leq \rho(\mathcal{E}) \Big(\sum_{i=1}^{J} \|\mathsf{u}_i\|_{\mathsf{B}}^2 \Big)^{1/2} \Big(\sum_{j=1}^{J} \|\mathsf{v}_j\|_{\mathsf{B}}^2 \Big)^{1/2}\end{aligned}$$

where $\rho(\mathcal{E}) = \|\mathcal{E}\|_2$ is the spectral radius of the matrix ε_{ij}. Note that the entries of the matrix ε_{ij} are zero for non overlapping subdomains. Indeed, suppose that $\text{supp}\{\mathrm{u}_i\} \subseteq \Omega_i$. Then $\text{supp}\{\mathrm{T}\mathrm{u}_i\} \subseteq \Omega_i$. This is a direct consequence of the use of broken test spaces, i.e., T, the DPG *trial-to-test operator* is local and therefore $\mathrm{T}\mathrm{u}_i$ is discontinuous. This results in

$$b_E(\mathrm{u}_i, \mathrm{v}_j) = b(\mathrm{u}_i, \mathrm{T}\mathrm{v}_j) = (\mathrm{T}\mathrm{u}_i, \mathrm{T}\mathrm{v}_j)_V = 0,$$

for all $\mathrm{v}_j \in \Omega_j$ such that $\text{supp}\{\mathrm{v}_j\} \cap \Omega_i = \emptyset$. Therefore

$$b_E(\mathrm{u}_i, \mathrm{v}_j) = 0,$$

for Ω_i, Ω_j disjoint and $\text{supp}\{\mathsf{u}_i\} \subseteq \Omega_i$, $\text{supp}\{\mathsf{v}_j\} \subseteq \Omega_j$. Consequently, by the finite overlap assumption (4.12), we obtain an upper bound for the spectral radius:

$$\rho(\mathcal{E}) \leq r$$

and the final result reads:

$$\sum_{i=1}^{J}\sum_{j=1}^{J} |b_E(\mathsf{u}_i, \mathsf{v}_j)| \leq r\Big(\sum_{i=1}^{J} \|\mathsf{u}_i\|_E^2\Big)^{1/2}\Big(\sum_{j=1}^{J} \|\mathsf{v}_j\|_E^2\Big)^{1/2}$$

4.3.4 Stable Decomposition

Proving the second assumption in Theorem 4.3 is a bit more involved. This is exactly the well-known *stable decomposition* assumption, which in simple terms it says that if u is decomposed into patch contributions, then the sum of energies stored in the patches must be controlled by the energy of u.

We start with a global stability result [19, 33],

$$\|\mathfrak{u}\|^2 + \|\hat{\mathfrak{u}}\|_{\hat{U}}^2 \leq \left[\frac{1}{\gamma^2} + (1+\frac{1}{\gamma})^2\right] \|(\mathfrak{u}, \hat{\mathfrak{u}})\|_E^2 \,.$$

Above γ denotes the global boundedness below constant for operator A. For the acoustics operator and the case of impedance BC, constant γ is independent of frequency ω [33]. Combining the stability estimate with continuity we derive the equivalence relation:

$$\gamma_1^2(\|\mathfrak{u}\|^2 + \|\hat{\mathfrak{u}}\|_{\hat{U}}^2) \leq \|(\mathfrak{u}, \hat{\mathfrak{u}})\|_E^2 \leq M_1^2(\|\mathfrak{u}\|^2 + \|\hat{\mathfrak{u}}\|_{\hat{U}}^2)$$

where $\gamma_1^{-2} = \frac{1}{\gamma^2} + (1+\frac{1}{\gamma})^2$

At the expense of having the constants γ_1 and M_1 in the estimate, we can construct a stable decomposition for the original trial norm instead of the energy norm. We will prove the result by considering separate cases for the L^2 space and the trace space $\hat{U}$. The stable decomposition for the L^2 space is standard but it is presented below for completeness.

4.3.4.1 Stable decomposition for the L^2 space

Let $Y_p = \mathcal{Q}^{(p,q)} = \mathcal{P}^p \otimes \mathcal{P}^q$ denote the space of polynomials of order less or equal p, q with respect to x, y respectively. These spaces are described in detail in (2.18). Additionally, let $\{\Phi_j\}_{j=1}^J$ be a partition of unity subordinate to the covering Ω_j of Ω, so that

$$\sum_{j=1}^{J} \Phi_j = 1, \qquad 0 \leq \Phi_j \leq 1, \qquad \text{supp}(\Phi_j) \subset \Omega_j$$

Given a $\mathfrak{u} \in Y_p$ we define the following decomposition

$$\mathfrak{u}_j = \mathsf{P}\Phi_j \mathfrak{u}$$

where $\mathsf{P} : \mathcal{Q}^{(p+1,q+1)} \to \mathcal{Q}^{(p,q)}$ is the L^2 orthogonal projection. Obviously

$$\sum_{i=1}^{J} \mathfrak{u}_j = \sum_{j=1}^{J} \mathsf{P}\Phi_j \mathfrak{u} = \mathsf{P} \underbrace{\sum_{i=1}^{J} \Phi_j}_{=1} \mathfrak{u}_j = \mathsf{P}\mathfrak{u} = \mathfrak{u}$$

It easy to see that the decomposition is stable. Indeed

$$\|\mathfrak{u}_j\| = \|\mathsf{P}\Phi_j \mathfrak{u}\| = \|\mathsf{P}\Phi_j \mathfrak{u}\|_{L^2(\Omega_j)} \leq \|\mathsf{P}\| \|\Phi_j\|_{L^\infty} \|\mathfrak{u}\|_{L^2(\Omega_j)} \leq \|\mathfrak{u}\|_{L^2(\Omega_j)}$$

Summing up for all subdomains, and by the finite overlap property (4.12) we have

$$\sum_{j=1}^{J} \|\mathfrak{u}_j\|^2 \leq \sum_{j=1}^{J} \|\mathfrak{u}\|^2_{L^2(\Omega_j)} \leq r\|\mathfrak{u}\|^2$$

We focus now on the construction of a stable decomposition for the trace spaces.

4.3.4.2 Stable decomposition for the trace space $\hat{U}$

We shall consider a single element $K \in \Omega_j$, see Figure 4.1 for an illustration. We denote by $H_A(K)$ the graph energy space of functions defined on element K. Recall that $\bar{\Omega}_j$ is defined by the support of a vertex shape function Φ_j of size δ. Then, $\{\Phi_j\}_{i=1}^J$ is a partition of unity,

$$\sum_{j=1}^{J} \Phi_j = 1, \qquad 0 \leq \Phi_j \leq 1, \qquad \text{supp}(\Phi_j) \subset \Omega_j \quad \text{and} \quad \|\nabla \Phi_j\|_{L^\infty} \leq C\delta^{-1}$$

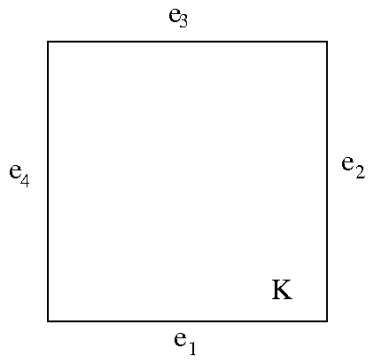


Figure 4.1: A single element in a vertex patch.

Operator dependent projection based interpolant. Let $\hat{\mathfrak{u}}$ be a sufficiently regular function defined on the skeleton. Additionally, let's assume the existence of a projection-based interpolant $\hat{\Pi}\hat{\mathfrak{u}}$ (which will be defined explicitly later on) taking values on the mesh skeleton such that

$$\|\hat{\Pi}\Phi_j\hat{\mathfrak{u}}\|_{\hat{U}} \leq \underbrace{\|\hat{\Pi}\|_{\hat{U}}}_{=:C_{\hat{\Pi}}} \|\Phi_j\hat{\mathfrak{u}}\|_{\hat{U}} \tag{4.13}$$

Construction of such an interpolant is operator-dependent, and we expect constant $C_{\hat{\Pi}}$ to grow mildly with the frequency ω for the acoustics operator, and be independent of the polynomial order p and the fine mesh size h. We will investigate the dependence on ω, h, and p numerically by constructing such an interpolation operator. The construction and the numerical experiments are described in the next section. Consider now a function on the skeleton $\hat{\mathfrak{u}}$. We define the corresponding decomposition by

$$\hat{\mathfrak{u}}_j = \hat{\Pi}\Phi_j\hat{\mathfrak{u}}\,.$$

Clearly,

$$\sum_{j=1}^{J}\hat{\mathfrak{u}}_j = \sum_{j=1}^{J}\hat{\Pi}\Phi_j\hat{\mathfrak{u}} = \hat{\Pi}(\underbrace{\sum_{j=1}^{J}\Phi_j\,\hat{\mathfrak{u}})}_{=1} = \hat{\Pi}\hat{\mathfrak{u}} = \hat{\mathfrak{u}}$$

and by the postulated property (4.13)

$$\|\hat{\mathfrak{u}}_j\|_{\hat{U}} = \|\hat{\Pi}\Phi_j\hat{\mathfrak{u}}\|_{\hat{U}} \leq C_{\hat{\Pi}}\|\Phi_j\hat{\mathfrak{u}}\|_{\hat{U}} \leq C_{\hat{\Pi}}\|\Phi_j\mathfrak{u}\|_{H_A}$$

where $\mathfrak{u}$ is an extension of $\hat{\mathfrak{u}}$ to the element such that $tr\mathfrak{u} = \hat{\mathfrak{u}}$. The last term bounds nicely for the acoustics operator,

$$\|\Phi_j \mathfrak{u}\|^2_{H_{\mathrm{A}}} = \|\mathrm{A}(\Phi_j \mathfrak{u}))\|^2 + \|(\Phi_j \mathfrak{u})\|^2$$

and

$$\begin{aligned}\|\mathrm{A}(\Phi_j(u,p))\|^2 &= \|i\omega\Phi_j p + \mathrm{div}(\Phi_j u)\|^2 + \|i\omega u + \nabla(\Phi_j p)\|^2 \\ &\leq 2\Big(\|\Phi_j(i\omega p + \mathrm{div}\ u)\|^2 + \|\Phi_j(i\omega u + \nabla p)\|^2 + \|\nabla\Phi_j \cdot u\|^2 + \|\nabla\Phi_j p\|^2\Big) \\ &\leq 2\Big(\|\mathrm{A}(u,p)\|^2 + \delta^{-2}(\|u\|^2 + \|p\|^2)\Big)\end{aligned}$$

Therefore

$$\|\hat{\mathfrak{u}}_j\|_{\hat{U}} \leq C_{\hat{\Pi}} C_\delta \|\mathfrak{u}\|_{H_A}, \quad \text{where} \quad C_\delta = \mathcal{O}(\delta^{-1})$$

Note that the constant C_δ is independent of frequency. The above inequality is true for an arbitrary extension $\mathfrak{u}$ of $\hat{\mathfrak{u}}$. Taking the infimum on the right hand side with respect to extensions and summing over all subdomains we obtain the stable decomposition for the trace space. Combining everything together we have the following lemma.

Lemma 4.5. *For every function $\mathsf{u} = (\mathfrak{u}, \hat{\mathfrak{u}}) \in \boldsymbol{L}^2(\Omega) \times \hat{U}$ there exists a stable decomposition*

$$\mathsf{u} = \sum_{j=1}^{J} \mathsf{u}_j, \quad \mathsf{u}_j \in \boldsymbol{L}^2(\Omega_j) \times \hat{U}_j,$$

such that

$$\sum_{j=1}^{J} \|\mathsf{u}_j\|^2_E \leq C\|\mathsf{u}\|^2_E\,.$$

where $C = \mathcal{O}(rC^2_{\hat{\Pi}} C^2_\delta)$. Here r comes from the finite overlap property and $C_\delta = \mathcal{O}(\delta^{-1})$. This in turn gives an upper bound for the condition number of the preconditioned DPG system

$$\kappa(\mathsf{AB}) \leq C\frac{r^2}{\delta^2}, \quad \textit{where} \quad C = \mathcal{O}(C^2_{\hat{\Pi}}). \tag{4.14}$$

It remains to study the dependence of the interpolation norm $C_{\hat{\Pi}}$ on the discretization size h, polynomial order p and the frequency ω. For this, we design a numerical experiment to compute the interpolation norm. A discussion on the construction and the numerical results is presented below.

4.3.5 Computing the interpolation norm

The continuity constant $C_{\hat{\Pi}} = \|\hat{\Pi}\|_{\hat{U}}$ can be computed using the definition of a norm of an operator, i.e., by solving the maximization problem:

$$\|\hat{\Pi}\|_{\hat{U}} = \max_{\hat{v}\in\hat{U},\hat{v}\neq 0} \frac{\|\hat{\Pi}\hat{v}\|_{\hat{U}}}{\|\hat{v}\|_{\hat{U}}}$$

This leads to the generalized eigenvalue problem

$$(\hat{\Pi}\hat{v}, \hat{\Pi}\widehat{\delta v})_{\hat{U}} = \lambda^2 (\hat{v}, \widehat{\delta v})_{\hat{U}}, \qquad \widehat{\delta v} \in \hat{U}.$$

Consider now a discrete basis $\{\hat{v}_i\}_{i=1}^n$ for the polynomial subspace $\hat{U}_h \subset \hat{U}$. Then, we need to solve

$$\mathsf{P}^*\mathsf{G}\mathsf{P}\mathsf{v} = \lambda^2 \mathsf{G}\mathsf{v} \tag{4.15}$$

where $\mathsf{v} \in \mathbb{C}^n$, P is the matrix representation of $\hat{\Pi}$ and $\mathsf{G} = (\hat{v}_i, \hat{v}_j)_{\hat{U}}$ is the Gram matrix corresponding to the inner product $(\cdot,\cdot)_{\hat{U}}$. The continuity constant is therefore given by the square root of the maximum generalized eigenvalue of (4.15).

Computation of matrix $\boldsymbol{G}$. Given a basis $\{\hat{v}_i\}_{i=1}^n$ of the polynomial subspace $\hat{U}_h \subset \hat{U}$ we can compute the entries of the Gram matrix by using the $(\cdot,\cdot)_{H_{\mathrm{A}}}$ inner product. Indeed, using the polarization formula [101, Ch. 6], it is easy to see that

$$(\hat{v}_i, \hat{v}_j)_{\hat{U}} = (v_i, v_j)_{H_A} \tag{4.16}$$

where $v_i, v_j \in H_{\mathrm{A}}(K)$ are the minimum energy extensions of $\hat{v}_i, \hat{v}_j$ respectively. However, in practice we can only approximate the minimum energy extensions using polynomial extensions. We note that for all the numerical results presented in Section 4.3.6 polynomials of sufficiently large order [2] were used.

[2] No notable change in the numerical results could be observed when increasing further the polynomial order of approximation

Approximating the minimum energy extension. Given $\hat{v} \in \hat{U}_h$ we can derive the extension $v \in H_A(K)$ that realizes the minimum energy by:

$$\|\hat{v}\|_{\hat{U}} = \inf_{trv=\hat{v}} \|v\|_{H_{\mathrm{A}}}$$

Consider the polynomial subspace $H_{\mathrm{A},h}(K) \subset H_{\mathrm{A}}(K)$. Then the above minimization problem leads to the following Dirichlet problem.

$$\begin{cases} \text{Find } v_h \in H_{\mathrm{A},h}(K) \\ (v_h, \delta v_h)_{H_{\mathrm{A}}} = 0, \quad \delta v_h \text{ in } H_{\mathrm{A},h}(K) \\ v_h = \hat{v}, \qquad\qquad\qquad \text{on } \partial K \end{cases}$$

or equivalently

$$\begin{cases} \text{Find } v_b \in H^b_{\mathrm{A},h}(K) \text{ (bubble functions)} \\ (\mathrm{A}v^b, \mathrm{A}\delta v^b) + (v^b, \delta v^b) = -(\hat{v}, \delta v^b)_{H_{\mathrm{A}}}, \quad \forall \delta v^b \text{ in } H^b_{\mathrm{A},h}(K) \end{cases}$$

Definition of the interpolation operator $\mathbf{\Pi}$ - computation of matrix $\mathbf{P}$. Let $\hat{v} \in \hat{U}_h(K)$. We define the projection based interpolant $\hat{v}^p = \hat{\Pi}\hat{v}$ using the following steps:

- **Interpolation at vertices.** The interpolant $\hat{v}^p$ matches the function $\hat{v}$ at the vertices

  $$\hat{v}^p(a) = \hat{v}(a), \qquad \forall \text{ vertex } a.$$

  We lift the vertex values using a polynomial extension that lives in the element trace space. This leads to the linear interpolant $\hat{v}_1 \in tr\mathcal{P}^1(K)$.

- **Edge projection.** We subtract the linear interpolant $\hat{v}_1$ from the function $\hat{v}$. Now the difference $\hat{v} - \hat{v}_1$ vanishes at the element vertices. Then we project the difference onto the trace space of edge polynomials of order p_e vanishing at the vertices (i.e, the edge bubbles $\mathcal{P}_0^{p_e}$), i.e.,

  $$\begin{cases} \hat{v}_{2,e} \in \mathcal{P}_0^{p_e} \\ \|\hat{v} - \hat{v}_1 - \hat{v}_{2,e}\|_{\hat{U}(e)} \to \min . \end{cases}$$

  Here, the norm $\|\cdot\|_{\hat{U}(e)}$ is defined by the inner product (4.16) on the edge. The edge interpolant is then the sum of the edge projections

  $$\hat{v}_2 = \sum_e \hat{v}_{2,e}.$$

The final interpolant is defined by the sum of the vertex and the edge interpolant

$$\hat{v}^p = \hat{v}_1 + \hat{v}_2.$$

We obtain a matrix representation of P by applying it to a basis of the polynomial space. Recall the polynomial spaces defined in (2.18). In particular, consider

$$W^r := \mathcal{Q}^{(r,r)} \subset H^1(\Omega)$$
$$V^r := \mathcal{Q}^{(r,r-1)} \times \mathcal{Q}^{(r-1,r)} \subset H(\mathrm{div}, \Omega).$$

Then

$$\hat{\Pi} : W^{r+1} \times V^{r+1} \to W^r \times V^r,$$

and its matrix representation P is computed as follows. For $\hat{p} \in \ tr|_{\partial K} W^{r+1}$, we have vertex shape functions and bubble shape functions. For the vertex shape functions the interpolation operator matches the values at the vertices and so the first 4 columns of matrix P are the orthonormal vectors $\{e_i\}_{i=1}^4$. Note that if $\hat{p}$ is an edge bubble then it vanishes on the vertices. Therefore it is enough to solve the following minimization problem. Let $\hat{p} \in tr|_{\partial K} \mathcal{W}_0^{r+1}$ be an edge bubble vanishing on the boundary. Then we solve

$$\begin{cases} \text{Find } \hat{v}^b \in tr|_{\partial K} \mathcal{W}_0^r \\ \|\hat{p} - \hat{v}^b\|_{\hat{U}} \to \min \end{cases}$$

or equivalently

$$\begin{cases} \text{Find } \hat{v}^b \in tr|_{\partial K} \mathcal{W}_0^r \\ (\hat{v}^b, \delta\hat{v}^b)_{\hat{U}} = (\hat{p}, \delta\hat{v}^b)_{\hat{U}} \qquad \forall \delta\hat{v}^b \in tr|_{\partial K} \mathcal{W}_0^r \end{cases}$$

or

$$\begin{cases} \text{Find } v^b \in \mathcal{W}_0^r \\ (v^b, \delta v^b)_{H_{\mathrm{A}}} = (p^b, \delta v^b)_{H_{\mathrm{A}}} \qquad \forall \delta v^b \in \mathcal{W}_0^r \end{cases} \tag{4.17}$$

where $p^b, v^b, \delta v^b$ are the minimum energy polynomial extensions of $\hat{p}, \hat{v}^b, \delta\hat{v}^b$ respectively. Finally the interpolant is defined to be the trace of the solution v^b. Notice that both left and right hand sides of (4.17) can be computed using the Gram matrix G. For the edge bubbles of the variable $\hat{u}_n \in tr|_{\partial K} V^{r+1}$, we follow a similar procedure.

Computation on the master element. We can reduce the computation on the master element using appropriate scalings. Assuming that each element K is obtained by a simple scaling of the master element $\bar{K}$, and using the same H^1 scaling for both H^1 and $H(\text{div})$ functions we have:

$$\begin{aligned}
\|\mathrm{A}_\omega(p,u)\|^2_{L^2(K)} :&= \int_K (|i\omega p + \text{div } u|^2 + |i\omega u + \nabla p|^2)\, dK \\
&= \int_K h^2(|i\omega \bar{p} + \frac{1}{h}\bar{\text{div}}\ \bar{u}|^2 + |i\omega \bar{u} + \frac{1}{h}\bar{\nabla}\bar{p}|^2)\, d\bar{K} \\
&= \int_K (|i\omega h\bar{p} + \bar{\text{div}}\ \bar{u}|^2 + |i\omega h\bar{u} + \bar{\nabla}\bar{p}|^2)\, d\bar{K} \\
&=: \|\mathrm{A}_{\omega h}(p,u)\|^2_{L^2(\bar{K})}
\end{aligned}$$

and

$$\|(p,u)\|^2_{L^2(K)} = h^2\|(\bar{p},\bar{u})\|^2_{L^2(\bar{K})}$$

We can therefore perform all the computations on the master element by using the following norm:

$$\|(p,u)\|^2_{H_{\mathrm{A}}(\omega h,h)} = \|\mathrm{A}_{\omega h}(p,u)\|^2_{L^2(\bar{K})} + h^2\|(\bar{p},\bar{u})\|^2_{L^2(\bar{K})}$$

4.3.6 Results

We examine the dependence of the norm of the interpolation operator described above, on the polynomial order p, the discretization size h, and the frequency ω. We present the results in the tables below for polynomial orders $p = 2, 4, 6$. For each case of p we also present convergence of the preconditioned CG solver. The subdomains are defined to be the support of a vertex function defined on the mesh of size $h = 1/2$ (i.e, 9 subdomains with a fixed overlap size $\delta = 1/2$. For these experiments we simulate a plane wave in the square domain $\Omega = (0,1)^2$, using impedance boundary conditions on the entire boundary of the domain. We run our simulations for five different frequencies and we perform successive uniform h-refinements, starting with a mesh of size $h = 1/2$. The CG solver is terminated when the l_2-norm of the residual drops below 10^{-6}. The numbers in red color denote that the error is above 90%, i.e, the mesh is not fine enough to resolve the wave.

h \ ω	π	2π	4π	8π	16π
1/2	15	16	17	11	8
1/4	16	18	20	14	9
1/8	16	18	22	23	10
1/16	16	18	23	24	25
1/32	16	19	23	25	25

(a) Iteration count

h \ ω	π	2π	4π	8π	16π
1/2	2.553	4.327	6.088	3.768	2.268
1/4	2.540	4.427	7.598	6.320	4.328
1/8	2.510	4.415	6.972	8.247	6.061
1/16	2.500	4.362	6.841	8.379	9.555
1/32	2.497	4.343	6.785	8.651	9.621

(b) Value of the constant $\|\hat{\Pi}\|_{\hat{U}}$

Table 4.1: Polynomial order $p = 2$. Left: iteration count for CG preconditioned with additive Schwarz smoother with fixed $\delta = 1/2$. Right: The value of the interpolation norm.

h \ ω	π	2π	4π	8π	16π
1/2	16	18	22	17	9
1/4	16	18	22	22	11
1/8	16	18	22	25	23
1/16	17	19	23	25	25
1/32	17	19	23	25	26

(a) Iteration count

h \ ω	π	2π	4π	8π	16π
1/2	1.516	1.976	3.222	3.168	1.871
1/4	1.497	2.021	3.328	5.262	2.582
1/8	1.491	2.046	3.432	5.122	5.891
1/16	1.490	2.055	3.492	5.260	6.478
1/32	1.489	2.057	3.512	5.352	6.556

(b) Value of the constant $\|\hat{\Pi}\|_{\hat{U}}$

Table 4.2: Polynomial order $p = 4$. Left: iteration count for CG preconditioned with additive Schwarz smoother with fixed $\delta = 1/2$. Right: The value of the interpolation norm.

h \ ω	π	2π	4π	8π	16π
1/2	17	19	23	24	12
1/4	16	18	22	23	22
1/8	17	18	22	24	24
1/16	17	18	22	25	25
1/32	17	18	23	25	26

(a) Iteration count

h \ ω	π	2π	4π	8π	16π
1/2	1.271	1.522	1.874	3.514	2.391
1/4	1.268	1.512	1.753	3.093	3.026
1/8	1.267	1.510	1.734	2.750	3.583
1/16	1.267	1.510	1.730	2.685	3.724
1/32	1.266	1.510	1.730	2.671	3.962

(b) Value of the constant $\|\hat{\Pi}\|_{\hat{U}}$

Table 4.3: Polynomial order $p = 6$. Left: iteration count for CG preconditioned with additive Schwarz smoother with fixed $\delta = 1/2$. Right: The value of the interpolation norm.

The following interesting observations can be derived from the results. First, the solver shows convergence, independent of the polynomial order. This, can also be verified in the value of the constant $\hat{\Pi}_{\hat{U}}$. Gopalakrishnan and Schöberl observed the same behavior in their construction of a multiplicative Schwarz preconditioner in [66]. The second observation involves the dependence on the mesh size h. As we can clearly see, both the constant and the CG convergence are independent of h.

Lastly, the number of iterations of the solver grows mildly with respect to the frequency, an observation that is also reflected on the value of the constant. However, in cases where the mesh is too coarse to resolve a high frequency wave, the solver is more efficient with respect to the number of iterations (see numbers in red). This behavior, might not seem to be of great value for uniform meshes, but in case of adaptive refinements, this could be very beneficial. As we have seen in the previous chapter, the DPG method being unconditionally stable, allows for adaptive refinements starting from very coarse meshes. Therefore, integrating an iterative solver within the adaptive process, even in the pre-asymptotic region, seems to be a promising direction to follow.

Overall, the computed interpolation norm gives a very useful insight on the behavior of the solver. We emphasize that, there is no reason to believe that the bound provided by the interpolation norm is sharp. However, the overall trend of the value of the norm, is consistent with the convergence behavior of the solver.

4.4 Extension to the multilevel setting

We would like to extend our construction to the two- (multi-) level setting. The main reason is the unavoidable dependence of the one-level preconditioner on the number of subdomains and the size of overlap. In order to, keep the number of iterations of the solver under control, both the number of subdomains and the size of the overlap have to remain constant and unfortunately this adds significant work on the local solves within each subdomain as h decreases. On the other hand, if the number of subdomains increases, the cost of each local solve remains constant, but the overall number of iterations grows.

Usually, convergence independent with respect to the number of subdomains and the size of the overlap can be achieved by adding a coarse grid correction step in the algorithm. The standard analysis involves treating the coarse grid solve as a local solve defined on an additional subspace (the coarse grid) and therefore condition number estimates can be derived using the subspace correction theory (see Theorem 4.3). In particular, comparison of the condition number estimates for the Poisson problem proven in [7] for the one-level setting and in [87] for the two-level setting, shows exactly the claim above. The result for the one- and two- level preconditioners are:

$$\kappa_{\text{one-level}} \leq C\frac{r^2}{\delta^2} \quad \text{and} \quad \kappa_{\text{two-level}} \leq Cr^2\Big(1+\frac{H^2}{\delta^2}\Big)$$

respectively. Here, H is the discretization size of the coarse grid, and C is independent of the fine grid discretization size h. Assuming that the number r that characterizes the finite overlap property remains constant (this is usually the case), then clearly for a coarse mesh of size $H = \mathcal{O}(\delta)$, the condition number of the two-level algorithm becomes independent with respect to the overlap.

For the case of the acoustics problem, it is much harder to get a similar estimate for the two-level setting. Combining our one-level setting result with a coarse grid solve in an additive way we expect the condition number to depend on the coarse grid discretization size H, the polynomial order p and the frequency ω. Although, we don't have a rigorous proof for this estimate, numerical experiments shown in Chapters 5 and 6 suggest that uniform convergence with respect to h, p and ω can be achieved when the coarse mesh is "fine enough" to capture the characteristics of the wave and overcome the pollution effect.

4.4.1 Additive vs multiplicative coupling

Schwarz type domain decomposition preconditioners are categorized into two main classes: the additive and the multiplicative. Both of them can be analyzed using the unified theory of subspace corrections of Xu [119, 118, 117]. Their main difference is the following: in the additive case the corrections are performed simultaneously on each subspace, but in

the multiplicative case they are carried out successively (each subspace solver operates on the updated residual).

Standard multigrid techniques are based on the multiplicative coupling of a coarse grid correction and an additive Schwarz solver, usually referred as the *smoother*. The additive smoother is usually preferred over the multiplicative one, because of its local nature. The solves on each subspace can be carried out in parallel and with today's multi-core computer architectures, computing times can be significantly reduced. On the other hand, coupling the smoother with a coarse grid solve in the multiplicative way can be shown to be much more effective than the alternative. This is called the *hybrid Schwarz method*. David Pardo et al. in [103, 104], demonstrated that the multiplicative coupling of a coarse grid correction and an additive smoother with an optimal relaxation parameter will always converge at a rate at least as fast as the additive coupling. In case of preconditioning, an improved estimate of the condition number for the multiplicative coupling is shown by Mandel in [89].

Chapter 5

A two grid preconditioner

In this chapter we describe the construction of a two-grid iteration scheme for the ultraweak DPG formulation. The construction is based on the multiplicative coupling of a coarse grid solve with an additive Schwarz smoother. This two-grid scheme is then used as a preconditioner for the conjugate gradient solver. We emphasize that the construction is for *hp*-meshes, i.e, meshes with hanging nodes and variable order. We then present results for the solution of the linear acoustics equations in both uniform and adaptive settings in two space dimensions [1]. We note that the solver is general, in the sense that it is not limited to wave problems. It can be applied to any DPG formulation, for any well posed problem involving the standard energy spaces.

Author contributions. The contents of this chapter are based on the published paper [106], co-authored by the author of this dissertation. The author of this dissertation contributed to the mathematical theory, software development and numerical simulations related to the work presented in [106].

5.1 Construction

We start by a general description on the construction. Recall that the DPG ultraweak formulation for the linear acoustics formulation involves L^2 field variables, $H^{1/2}$ variables and $H^{-1/2}$ variables. The L^2 variables have no continuity requirements among elements and so they are locally (element-wise) condensed out. Hence, the final system consists only of the interface

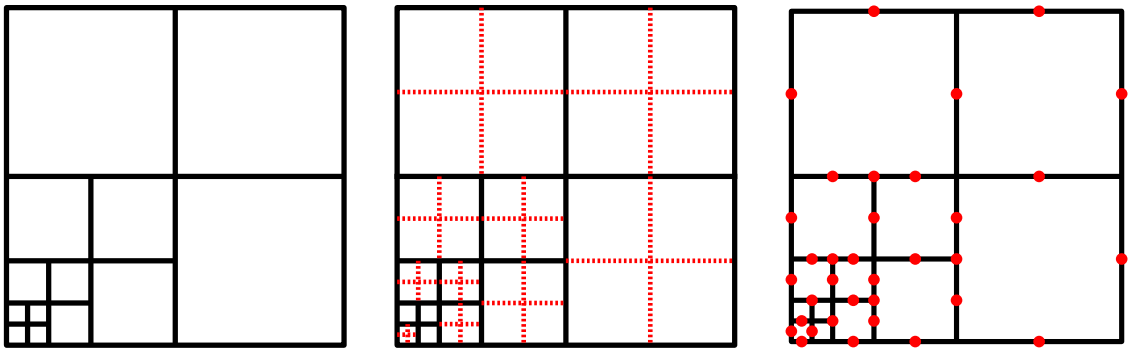
[1] The contents of this chapter are partially taken from the published paper: *Petrides, S. and Demkowicz, L. F. (2017). An adaptive DPG method for high frequency time-harmonic wave propagation problems. Comput. Math. Appl., 74(8):1999–2017.*

unknowns which are discretized by taking the appropriate trace of variables in the polynomial subspaces of H^1 and $H(\text{div})$ (see 2.18). We therefore need to construct a prolongation and a smoother operator for these trace spaces. An outline of this procedure is presented below.

Coarse grid correction. In general, the coarse grid correction step requires the construction of a prolongation operator which transfers a solution vector from the coarse to the fine mesh, and a restriction operator which restricts the residual vector on the fine grid to the coarse grid. In other words, the prolongation maps a coarse basis function to its representation in terms of the fine basis functions. While such a construction in the standard Galerkin method is straightforward, in the case of DPG it is a bit more complicated. The discretization, new edges (faces in 3D) are created after an h-refinement, therefore new interface variables appear, which have no ancestors. Consequently, the usual prolongation operator (natural inclusion) based on constrained approximation [37], would not be well defined in this case. In order to overcome this difficulty, we introduce the concept of a *macro-element* described in the next paragraph. The goal is to introduce a new mesh, which will have the same topology as the coarse mesh and so the prolongation operator will be easily defined.

In case of p-refinements with hierarchical shape functions, the new degrees of freedom are simply set to zero. This is the standard inclusion operator for the p multigrid algorithm. Finally, the restriction operator is defined to be the transpose of the prolongation operator.

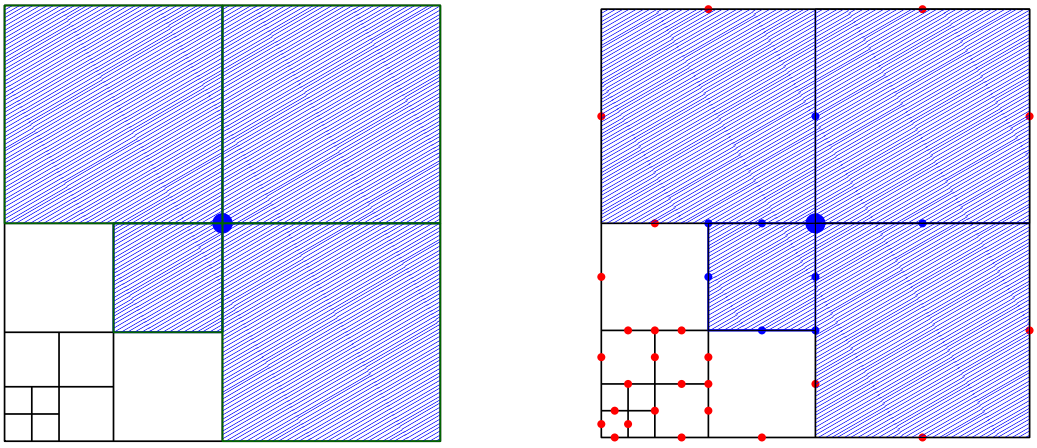
Macro-element. Consider a coarse and a fine grid (see Figure 5.1.) Suppose that the fine grid is the mesh produced after several adaptive hp-refinements applied to the coarse grid, and it is the current mesh where we seek the solution to the problem. We define the *macro* grid to be the resulting mesh after we condense out all the new degrees of freedom which do not lie on the skeleton of the coarse mesh. Notice that now the two meshes have the same topology. Practically, the construction of the prolongation operator reduces to a one (two in 3D) dimensional interpolation problem (see [25]).



(a) Coarse grid (b) Fine grid (c) Macro grid

Figure 5.1: Macro Grid Definition. The degrees of freedom on the fine grid that do not lie on the skeleton of the coarse grid are eliminated using Schur complements.

Additive Schwarz smoother. We use the standard additive Schwarz method for smoothing the residual on the macro-grid, i.e., a block Jacobi iteration scheme with overlapping blocks. We define a patch to be the support of a coarse grid vertex basis function (see Figure 5.2). A matrix block is then constructed by the interaction of the macro degrees of freedom within a patch. The additive smoother is preferred over the multiplicative one (Gauss Seidel with overlapping blocks) because of its local properties, i.e., the construction, the inversion of each individual matrix block and the action of the smoother in the residual can be implemented in parallel.



(a) Vertex patch - coarse grid (b) Vertex patch - macro grid

Figure 5.2: Construction of a smoother patch. A smoother patch is defined by the support of a coarse grid vertex basis function.

Symmetric two-grid cycle. We implement the symmetric two-grid cycle between the coarse mesh and the macro mesh (see Fig. 5.3). Let $\mathsf{Ax} = \mathsf{b}$ denote the linear system to be solved on the macro grid and let $\mathsf{x_n}$ be the solution to the n^{th} iteration of the CG algorithm. Additionally, let $\mathrm{I}_{\mathrm{C}}^{\mathrm{M}}$ denote the prolongation operator between the coarse and the macro grid. The restriction operator is defined by the transpose, $\mathrm{I}_{\mathrm{M}}^{\mathrm{C}} = (\mathrm{I}_{\mathrm{C}}^{\mathrm{M}})^*$, and the coarse grid correction operator is given by $\mathsf{Q} = \mathrm{I}_{\mathrm{M}}^{\mathrm{C}}\mathsf{A}_{\mathrm{C}}^{-1}(\mathrm{I}_{\mathrm{C}}^{\mathrm{M}})^*$, where $\mathsf{A}_{\mathrm{C}}^{-1}$ denotes the exact inverse of the global stiffness matrix at the coarse level.

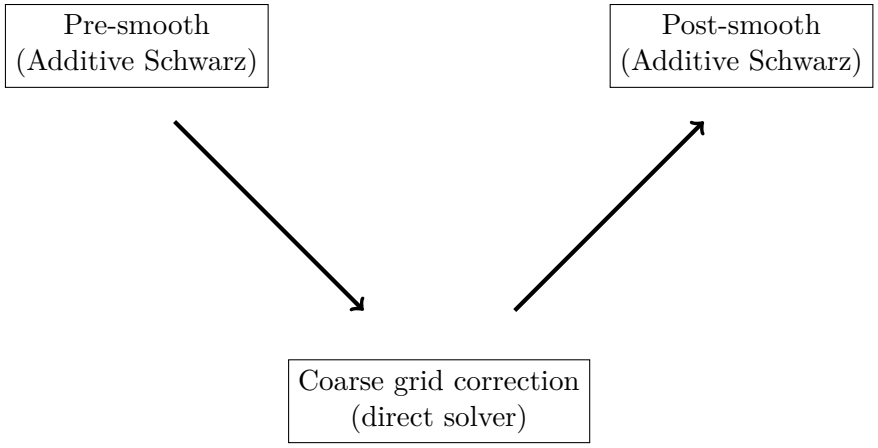


Figure 5.3: Two-grid cycle

Finally, for the smoothing procedure, we perform μ smoothing steps and use a damping parameter θ, where $0 < \theta < 1$ (θ is selected according to the finite overlap property (see (4.12)). We denote by S_μ the operator that performs μ smoothing steps of the additive Schwarz smoother. Note that S_μ is given by:

$$\mathsf{S}_\mu = \theta\mathsf{S}_1 \sum_{i=0}^{\mu-1} (\mathrm{I} - \theta\mathsf{A}\mathsf{S}_1)^i \tag{5.1}$$

where I is the identity operator and S_1 is the operator of a single smoothing step. Indeed, let $\mathsf{z}_\mathsf{n}^{\mathsf{s},i}$ be the i^{th} correction to solution, i.e, the correction after applying a single smoothing step to the i^{th} residual r_n^i, where $\mathsf{r}_\mathsf{n}^1 = \mathsf{r}_\mathsf{n}$. Then $\mathsf{z}_\mathsf{n}^\mathsf{s}$ is the sum of all the corrections:

$$\mathsf{z}_\mathsf{n}^\mathsf{s} = \sum_{i=1}^{\mu} \mathsf{z}_\mathsf{n}^{\mathsf{s},i} = \theta\mathsf{S}_1 \sum_{i=1}^{\mu} \mathsf{r}_\mathsf{n}^i = \theta\mathsf{S}_1 \sum_{i=1}^{\mu} (\mathrm{I} - \theta\mathsf{A}\mathsf{S}_1)^{i-1}\mathsf{r}_\mathsf{n}^1 = \sum_{i=0}^{\mu-1} (\mathrm{I} - \theta\mathsf{A}\mathsf{S}_1)^i \mathsf{r}_\mathsf{n}$$

and (5.1) immediately follows. The two-grid cycle algorithm is given below.

Algorithm 2 Two grid cycle

1: **procedure** TWOGRID($\mathsf{r_n}, \mathsf{z_n}$) ▷ Given $\mathsf{r_n}$ compute $\mathsf{z_n}$
2: $\mathsf{r_n} = \mathsf{b} - \mathsf{A}\,\mathsf{x_n}$ ▷ Compute initial residual
3: $\mathsf{z_n^s} = \mathsf{S}_\mu\,\mathsf{r_n}$ ▷ Pre-smooth μ times
4: $\mathsf{r_n^s} = \mathsf{r_n} - \mathsf{A}\mathsf{z_n^s} = (\mathrm{I} - \mathsf{A}\mathsf{S}_\mu)\mathsf{r_n}$ ▷ Update residual
5: $\mathsf{z_n^{qs}} = \mathrm{I}_{\mathrm{C}}^{\mathrm{M}}\,\mathsf{A}_{\mathrm{C}}^{-1}\,\mathrm{I}_{\mathrm{M}}^{\mathrm{C}}\,\mathsf{r_n^s} = \mathsf{Q}\,\mathsf{r_n^s}$ ▷ Coarse grid correction
6: $\mathsf{r_n^{qs}} = \mathsf{r_n^s} - \mathsf{A}\mathsf{z_n^{qs}} = (\mathrm{I} - \mathsf{A}\mathsf{Q})\mathsf{r_n^s}$ ▷ Update residual
7: $\mathsf{z_n^{sqs}} = \mathsf{S}_\mu\,\mathsf{r_n^{qs}}$ ▷ Post-smooth μ times
8: $\mathsf{z_n} = \mathsf{z_n^s} + \mathsf{z_n^{qs}} + \mathsf{z_n^{sqs}}$ ▷ Compute the total correction

Preconditioned Conjugate Gradient (PCG). Suppose that we denote by M the operator described in Algorithm 2. Note that if we want to use M as a preconditioner for the CG solver, then it has to be self-adjoint and positive definite. Positive definiteness comes by construction, but to ensure self-adjointness we have to perform equal number of pre- and post- smoothing steps. An explicit formula for M can be then derived by:

$$\begin{aligned}
\mathsf{z_n} &= \mathsf{z_n^s} + \mathsf{z_n^{qs}} + \mathsf{z_n^{sqs}} = \mathsf{S}_\mu\mathsf{r_n} + \mathsf{Q}\mathsf{r_n^s} + \mathsf{S}_\mu\mathsf{r_n^{qs}} \\
&= \mathsf{S}_\mu\mathsf{r_n} + \mathsf{Q}(\mathsf{b} - \mathsf{A}(\mathsf{x_n} + \mathsf{z_n^s})) + \mathsf{S}_\mu(\mathsf{b} - \mathsf{A}(\mathsf{x_n} + \mathsf{z_n^s} + \mathsf{z_n^{qs}})) \\
&= \mathsf{S}_\mu\mathsf{r_n} + \mathsf{Q}(\mathsf{r_n} - \mathsf{A}\mathsf{z_n^s}) + \mathsf{S}_\mu(\mathsf{r_n} - \mathsf{A}\mathsf{z_n^s} - \mathsf{A}\mathsf{z_n^{qs}}) \\
&= \mathsf{S}_\mu\mathsf{r_n} + \mathsf{Q}\mathsf{r_n} - \mathsf{Q}\mathsf{A}\mathsf{S}_\mu\mathsf{r_n} + \mathsf{S}_\mu\mathsf{r_n} - \mathsf{S}_\mu\mathsf{A}\mathsf{S}_\mu\mathsf{r_n} - \mathsf{S}_\mu\mathsf{A}\mathsf{Q}\mathsf{r_n^s} \\
&= \mathsf{S}_\mu\mathsf{r_n} + \mathsf{Q}\mathsf{r_n} - \mathsf{Q}\mathsf{A}\mathsf{S}_\mu\mathsf{r_n} + \mathsf{S}_\mu\mathsf{r_n} - \mathsf{S}_\mu\mathsf{A}\mathsf{S}_\mu\mathsf{r_n} - \mathsf{S}_\mu\mathsf{A}\mathsf{Q}(\mathsf{r_n} - \mathsf{A}\mathsf{S}_\mu\mathsf{r_n}) \\
&= (\mathsf{Q} + \mathsf{S}_\mu(\mathrm{I} - \mathsf{A}\mathsf{Q}) + (\mathrm{I} - \mathsf{Q}\mathsf{A})\mathsf{S}_\mu - \mathsf{S}_\mu(\mathrm{I} - \mathsf{A}\mathsf{Q})\mathsf{A}\mathsf{S}_\mu)\mathsf{r_n}.
\end{aligned}$$

Therefore, the preconditioner is given by

$$\mathsf{M} = \mathsf{Q} + \mathsf{S}_\mu(\mathrm{I} - \mathsf{A}\mathsf{Q}) + (\mathrm{I} - \mathsf{Q}\mathsf{A})\mathsf{S}_\mu - \mathsf{S}_\mu(\mathrm{I} - \mathsf{A}\mathsf{Q})\mathsf{A}\mathsf{S}_\mu \tag{5.2}$$

and the reduction of the residual is given by:

$$\begin{aligned}
\mathsf{r_n^{sqs}} &= \mathsf{b} - \mathsf{A}(\mathsf{x_n} + \mathsf{z_n^s} + \mathsf{z_n^{qs}} + \mathsf{z_n^{sqs}}) = \mathsf{r_n^{qs}} - \mathsf{A}\mathsf{z_n^{sqs}} \\
&= (\mathrm{I} - \mathsf{A}\mathsf{S}_\mu)\mathsf{r_n^{qs}} = (\mathrm{I} - \mathsf{A}\mathsf{S}_\mu)(\mathrm{I} - \mathsf{A}\mathsf{Q})\mathsf{r_n^s} \\
&= (\mathrm{I} - \mathsf{A}\mathsf{S}_\mu)(\mathrm{I} - \mathsf{A}\mathsf{Q})(\mathrm{I} - \mathsf{A}\mathsf{S}_\mu)\mathsf{r_n}, \\
&= (\mathrm{I} - \theta\mathsf{A}\mathsf{S}_1)^\mu(\mathrm{I} - \mathsf{A}\mathsf{Q})(\mathrm{I} - \theta\mathsf{A}\mathsf{S}_1)^\mu\mathsf{r_n}
\end{aligned}$$

where the last equality follows by equation (5.1). Notice how the multiplicative coupling becomes clear in the reduction of the residual. Finally, the preconditioned Conjugate Gradient algorithm [110] is given by Algorithm 3.

Algorithm 3 Preconditioned Conjugate Gradient

procedure PCG($\mathsf{x}_0, \mathsf{x}_n$) ▷ Given x_0 return x_n
  $\mathsf{r}_0 = \mathsf{b} - \mathsf{A}\mathsf{x}_0$
  $\mathsf{z}_0 = \mathsf{M}\mathsf{r}_0$
  $\mathsf{p}_0 = \mathsf{x}_0$
  for $\mathsf{j} = 1,2\ldots$ until convergence:
    $\alpha_j = \frac{(\mathsf{r}_j, \mathsf{z}_j)}{(\mathsf{A}\mathsf{p}_j, \mathsf{p}_j)}$
    $\mathsf{x}_{j+1} = \mathsf{x}_j + \alpha_j \mathsf{p}_j$
    $\mathsf{r}_{j+1} = \mathsf{r}_j - \alpha_j \mathsf{A}\mathsf{p}_j$
    $\mathsf{z}_{j+1} = \mathsf{M}\mathsf{r}_{j+1}$
    $\beta_j = \frac{(\mathsf{r}_{j+1}, \mathsf{z}_{j+1})}{(\mathsf{r}_j, \mathsf{z}_j)}$
    $\mathsf{p}_{j+1} = \mathsf{z}_{j+1} + \beta_j \mathsf{p}_j$
  end for

We test our preconditioner (5.2) to solve the 2D linear acoustics problem for various values of the frequency ω. We start in the uniform refinements setting and compare the two-grid preconditioner with the one-level additive Schwarz preconditioner. We proceed then to examples involving adaptive refinements. The results are summarized in the next section.

5.2 Smoother vs two grid preconditioner: uniform refinements

Our first experiment, is to compare the convergence of the preconditioned conjugate gradient solver with and without the coarse grid correction. We use the smoother described in the previous section and study the dependence on the frequency ω and the size of the coarse grid H (uniform refinements setting), and the polynomial order p.

5.2.1 Set up

We solve the linear acoustics equations on the square domain $\Omega = (0,1)^2$. The problem is driven by impedance boundary data read from the exact solution, a plane wave propagating

from the origin in a 45° angle. The ultraweak formulation is:

$$\begin{cases} u \in (L^2(\Omega))^d,\ p \in L^2(\Omega) \\ \hat{u}_n \in H^{-1/2}(\Gamma_h),\ \hat{p} \in H^{1/2}(\Gamma_h) \\ \hat{p} - \hat{u}_n = g,\ \text{on } \partial\Omega \\ (i\omega u, v) - (p, \text{div}_h v) + \langle \hat{p}, v \cdot n \rangle_{\Gamma_h} = 0, \quad v \in H(\text{div}, \Omega_h) \\ (i\omega p, q) - (u, \nabla_h q) + \langle \hat{u}_n, q \rangle_{\Gamma_h} \quad = 0, \quad q \in H^1(\Omega_h) \end{cases}$$

We run our simulations for polynomial orders $p = 2, 4, 6$ and for frequencies $\omega = \pi, 2\pi, \ldots, 16\pi$. In each case we perform four uniform h-refinements starting from a mesh of size $h = 1/2$. For each of these (fine) meshes, a coarse mesh is obtained by one uniform h-coarsening. A smoother patch (subdomain) is then defined to be the support of a coarse vertex shape function. It is important to note that this is not the same setting as in Section 4.3.6 where the overlap δ and the number of subdomains were fixed. On the contrary, here, after an h-refinement a new coarse grid is defined and therefore the number of the subdomains increases and the size of the overlap decreases (see Table 5.1). Note that this way, the cost of each local solve is constant because the size of each local problem remains the same.

h	δ	# Subdomains
1/2	1	4
1/4	1/2	9
1/8	1/4	25
1/16	1/8	36
1/32	1/16	49

Table 5.1: Overlap size vs h

For each run we use a relaxation parameter $\theta = 0.49$ (the finite overlap property suggests 1/2), and total of two smoothing iterations. The two grid preconditioner requires at least two smoother iterations, one pre-smooth and one post-smooth in order to remain self-adjoint. In order then to have a fair comparison, two smoothing iterations are used in the additive Schwarz preconditioner as well.

5.2.2 Results

A comparison of the two preconditioners is showcased in Tables 5.2 to 5.4. The tables on left show the convergence of the CG solver preconditioned with the two-grid strategy and on the right with the additive Schwarz smoother. Note that the numbers in red color indicate that the error is more than 90%. Our first observation suggests clear dependence of the smoother on the size of the overlap. Notice that for a fixed frequency the number of iterations grows (linearly) with the inverse of the overlap size (δ^{-1}). This is consistent with the bound (4.14) proved in the previous chapter. Secondly, we can clearly verify that the convergence of the smoother is independent with respect to the polynomial order.

h \ ω	π	2π	4π	8π	16π
1/2	5	6	5	3	3
1/4	5	7	16	10	4
1/8	5	8	12	30	12
1/16	5	7	13	24	51
1/32	5	7	14	26	47

(a) Preconditioner: two-grid. CG iterations

h \ ω	π	2π	4π	8π	16π
1/2	6	6	6	5	3
1/4	9	11	12	7	4
1/8	16	17	22	24	8
1/16	33	33	42	49	51
1/32	60	64	84	102	115

(b) Preconditioner: smoother. CG iterations

Table 5.2: Comparison of CG iteration count when preconditioned with two grid (left) and additive Schwarz (right) for $p = 2$

h \ ω	π	2π	4π	8π	16π
1/2	5	6	8	5	3
1/4	4	6	9	13	9
1/8	4	6	8	15	28
1/16	5	6	8	15	36
1/32	5	6	8	15	35

(a) Preconditioner: two-grid. CG iterations

h \ ω	π	2π	4π	8π	16π
1/2	6	7	6	5	3
1/4	9	10	13	13	9
1/8	17	17	21	25	27
1/16	33	33	42	48	55
1/32	60	64	78	105	113

(b) Preconditioner: smoother. CG iterations

Table 5.3: Comparison of CG iteration count when preconditioned with two grid (left) and additive Schwarz (right) for $p = 4$

h \ω	π	2π	4π	8π	16π
1/2	4	6	7	6	4
1/4	4	5	8	13	12
1/8	4	6	8	16	27
1/16	4	6	8	17	30
1/32	4	6	8	18	29

(a) Preconditioner: two-grid. CG iterations

h \ω	π	2π	4π	8π	16π
1/2	6	7	6	6	4
1/4	9	10	13	13	12
1/8	17	17	21	25	27
1/16	33	33	42	49	56
1/32	60	64	84	105	113

(b) Preconditioner: smoother. CG iterations

Table 5.4: Comparison of CG iteration count when preconditioned with two grid (left) and additive Schwarz (right) for $p = 6$

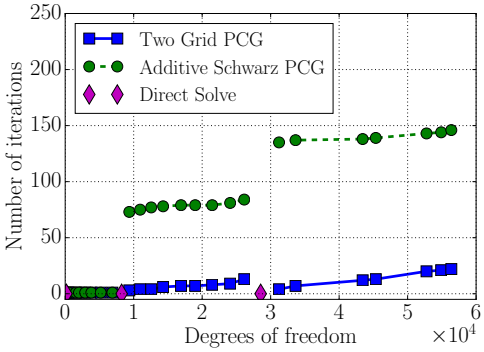
For the two-grid preconditioner we have the following observation. In cases where the coarse mesh is fine enough to resolve the wave and the pollution effect, the coarse grid correction offers significant acceleration of the solver. In fact, the convergence is then independent of the overlap size and the number of subdomains. Finally, notice that the two-grid preconditioner is always superior to the additive smoother. This means that even in cases where the coarse grid is not accelerating the convergence, it doesn't negatively affect it either.

5.3 Integrating the iterative solver with adaptivity - smoother vs two grid

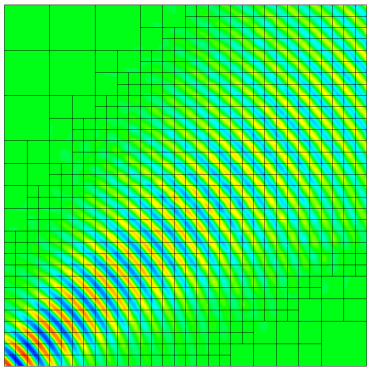
We would like to use the two grid technology in the adaptive refinement setting. Recall that the DPG method is unconditionally stable and therefore adaptive refinements can be initiated starting from very coarse meshes (see Section 3.4). Ideally, we can replace the direct solver in several adaptive refinements, with the CG solver preconditioned with the two grid scheme. Adaptivity can be driven by partially converged solutions, obtained by the iterative solver. Additionally, the same coarse grid can be used for several upcoming refinements and therefore the Cholesky factorization can be performed once, stored and used in the coarse grid solve step for several meshes. Note that this step would now involve only a triangular solve (back substitution). Finally the convergence can be slightly accelerated if the solution of a previous mesh is provided as an initial guess to the iterative solver.

5.3.1 High frequency Gaussian beam in free space

We run the two preconditioners for the same set up as in Section 3.4 for three different frequencies ($\omega = 40\pi$, 80π, 120π) and we summarize our results in Figures 5.4, 5.5 and 5.6 respectively. As a stopping criterion for CG iterations we use the discrete L^2 norm of the discrete residual. Since we are interested only in a partially converged solution (enough to perform meaningful refinements), a tolerance of 10^{-3} was used. Additionally, we perform several smoothing steps ($\mu = 10$), and use a damping parameter ($\theta = 0.49$). We run all the simulations until the L^2 relative error of the DPG method reduces below 10% when, at this point, the wave is resolved.

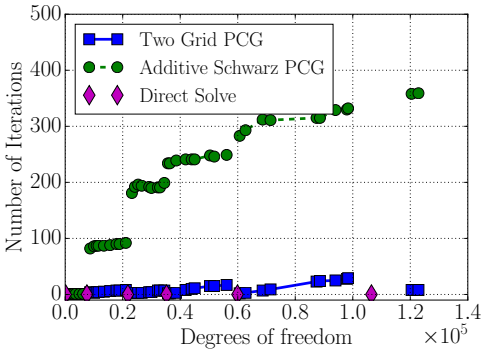


(a) Iteration count vs dof

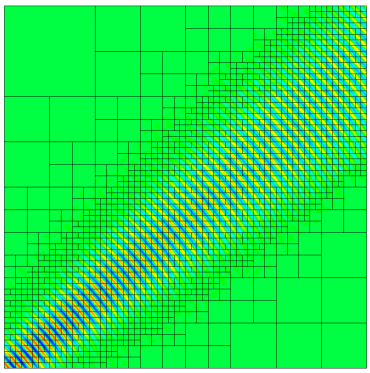


(b) Numerical pressure (real part)

Figure 5.4: Convergence of the PCG solver for $\omega = 40\pi$

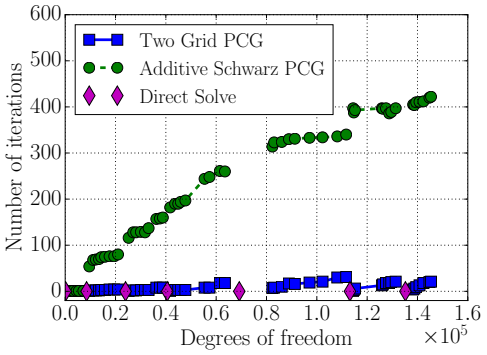


(a) Iteration count vs dof

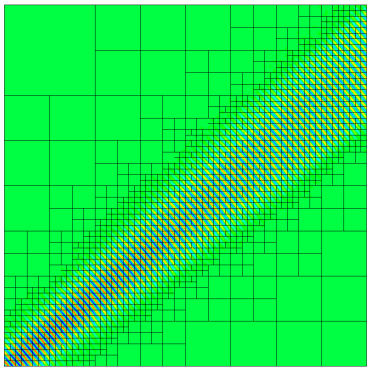


(b) Numerical pressure (real part)

Figure 5.5: Convergence of the PCG solver for $\omega = 80\pi$



(a) Iteration count vs dof



(b) Numerical pressure (real part)

Figure 5.6: Convergence of the PCG solver for $\omega = 120\pi$

Figures 5.4b, 5.5b and 5.6b show the final mesh and the pressure component of the solution. Clearly, we can conclude that the tolerance used was small enough to produce optimal adaptive refinements. Additionally, in Figures 5.4a, 5.5a and 5.6a, we compare the two preconditioners, the additive Schwarz and the two-grid described in the previous section. For the two-grid preconditioner, we follow a simple strategy, where the coarse grid is redefined every 10 refinements. Although the additive Schwarz converges relatively fast, the number of iterations grows every time the coarse grid is reset, i.e., the patches are redefined. The coarse grid correction seems to be necessary in order for the number of iterations to remain bounded, and that makes the two-grid strategy superior. The number of iterations drops every time we reset the coarse grid, and starts growing very slowly as we proceed with refinements. Intuitively, this is expected because the coarse grid correction becomes less effective, since the macro grid increasingly differs from the coarse grid as we keep refining. However, the number of iterations needed until convergence for the two-grid preconditioner appears to be independent of the frequency and the mesh.

5.3.2 High frequency Gaussian beam scattering by a cavity

We also test the proposed solver on a problem where we do not have the exact solution. Consider the domain of Figure 5.7. In this case we use the Gaussian beam as a source and

simulate the scattering of the wave from a resonating cavity. The problem is driven by an impedance boundary condition on Γ_2. On Γ_1 we put hard boundary condition $u \cdot n = 0$.

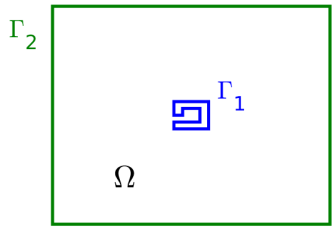


Figure 5.7: Computational domain containing a cavity

The ultraweak formulation for this setup is:

$$\begin{cases} u \in (L^2(\Omega))^d,\ p \in L^2(\Omega) \\ \hat{u}_n \in H^{-1/2}(\Gamma_h),\ \hat{p} \in H^{1/2}(\Gamma_h) \\ \hat{u}_n = 0, \text{ on } \Gamma_1, \quad \hat{p} - \hat{u}_n = g, \text{ on } \Gamma_2 \\ (i\omega u, v) - (p, \mathrm{div}_h v) + \langle \hat{p}, v \cdot n \rangle_{\Gamma_h} = 0, \quad v \in H(\mathrm{div}, \Omega_h) \\ (i\omega p, q) - (u, \nabla_h q) + \langle \hat{u}_n, q \rangle_{\Gamma_h} \quad = 0, \quad q \in H^1(\Omega_h) \end{cases}$$

We solve the problem for frequency $\omega = 1500\pi$. As a rule of thumb, standard methods for such a frequency need about four elements of polynomial order three per wavelength in order to become stable and produce meaningful solutions. For our example, this would result in 4000 elements in each direction or a total of sixteen million elements. However, the DPG discrete stability allows to start the simulation with a uniform mesh that only captures the geometry of the cavity. Our initial mesh consists of approximately 1000 cubic elements. This is obviously a very coarse mesh, with respect to the frequency of the problem. We also use a marking strategy to deal with the singularities at the corners of the cavity, i.e., we force h-refinements for elements adjacent to the corners. In Figure 5.8 and Figure 5.9 we show the evolution of the mesh and the solution respectively.

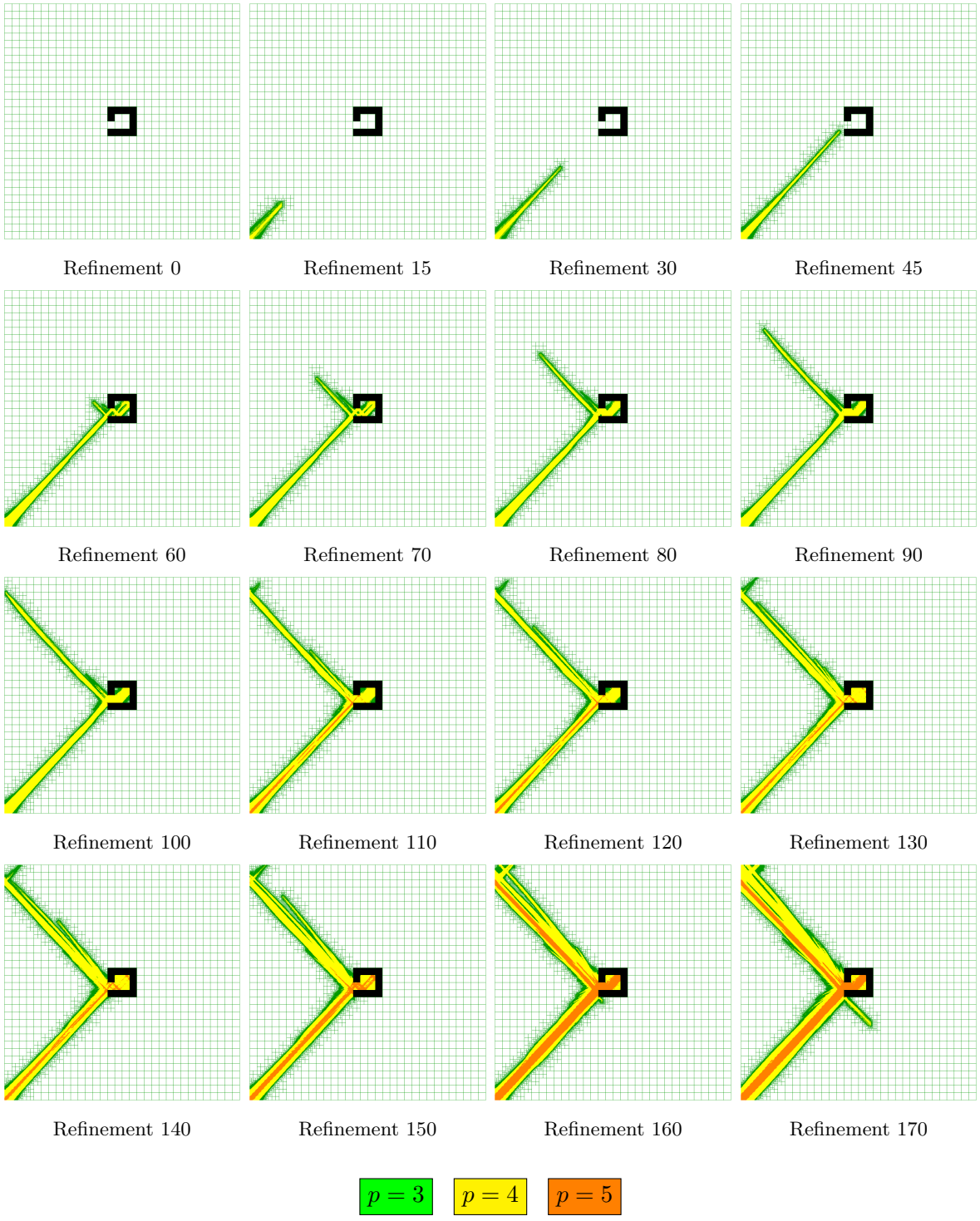


Figure 5.8: Adaptive *hp*-refinements for $\omega = 1500\pi$. Notice how the mesh is built along with the solution.

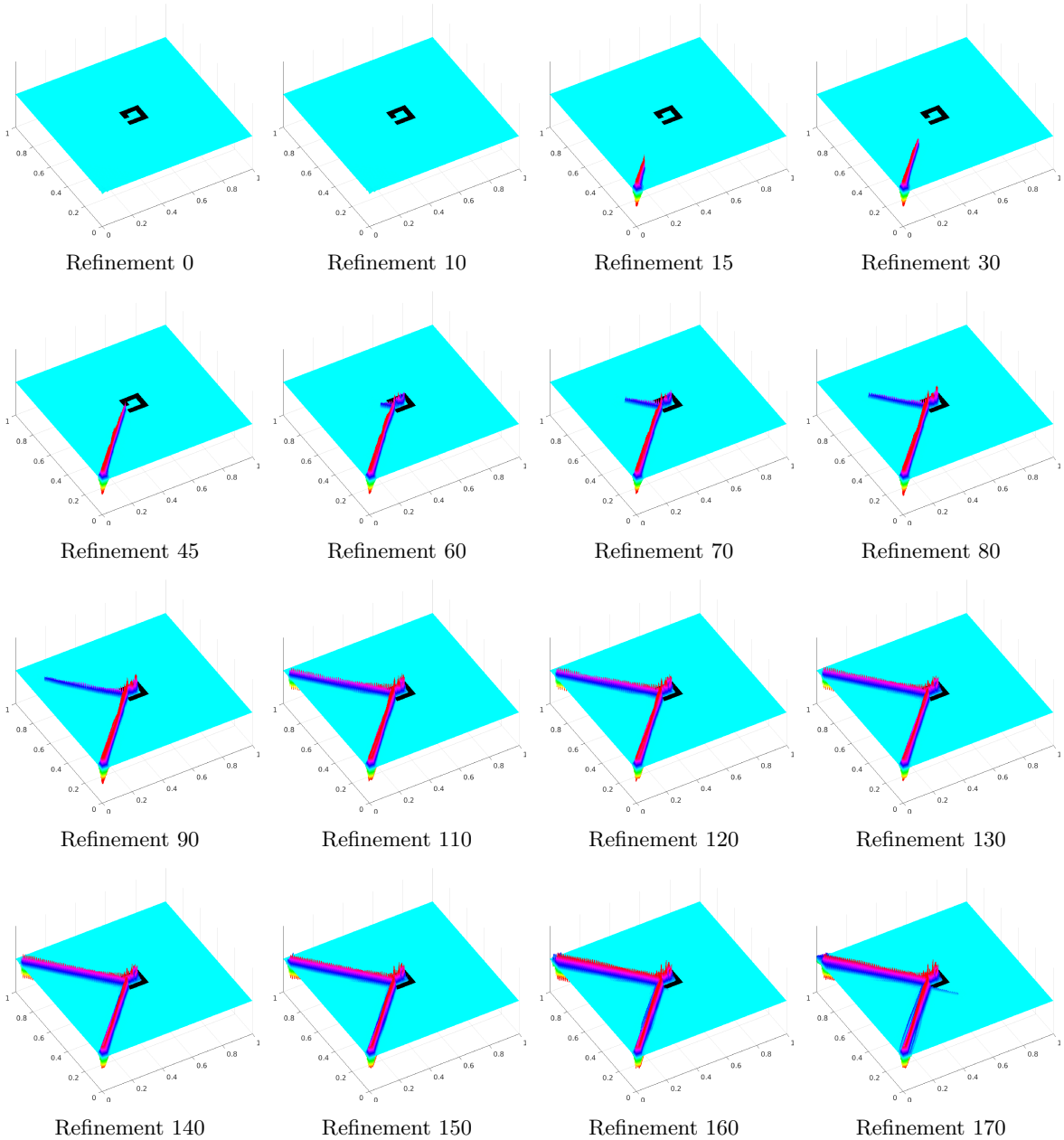


Figure 5.9: Real part of numerical acoustic pressure for $\omega = 1500\pi$

As we can see, the refinements are optimally carried out by the partially converged solution from the iterative solver. This can be verified also by Section 5.3.2. In this figure we show the convergence of the residual with respect to the skeleton degrees of freedom for two cases: a) the two-grid PCG solver and b) a direct solver. As it is clear from the plot, both solvers produce almost identical refinement patterns, i.e, in the end they deliver the same mesh for the same residual.

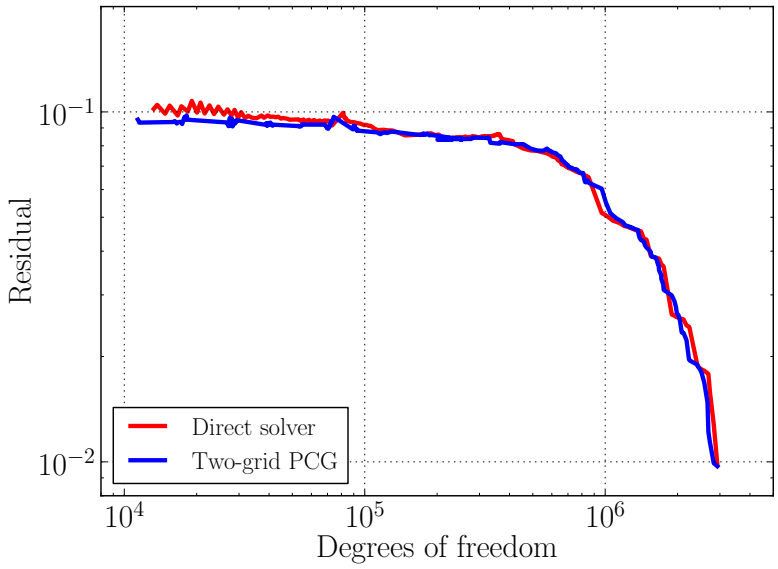


Figure 5.10: PCG vs direct solver for $\omega = 1500\pi$: residual vs skeleton dof. As demonstrated the two solvers produce almost identical refinement patterns. Therefore, adaptivity can indeed be driven by partially converged solutions.

Additionally, in Figure 5.11 we compare the number of CG iterations, when preconditioned with the additive Schwarz preconditioner and the two-grid preconditioner. The behavior of the two preconditioners is the same as in the previous examples. The superiority of the two-grid solver is apparent. The two-grid PCG solver always converges in less than 20 iterations. On the other hand the convergence of the additive Schwarz PCG, as expected, worsens when the number of subdomains (smoother patches) grows and the overlap becomes smaller.

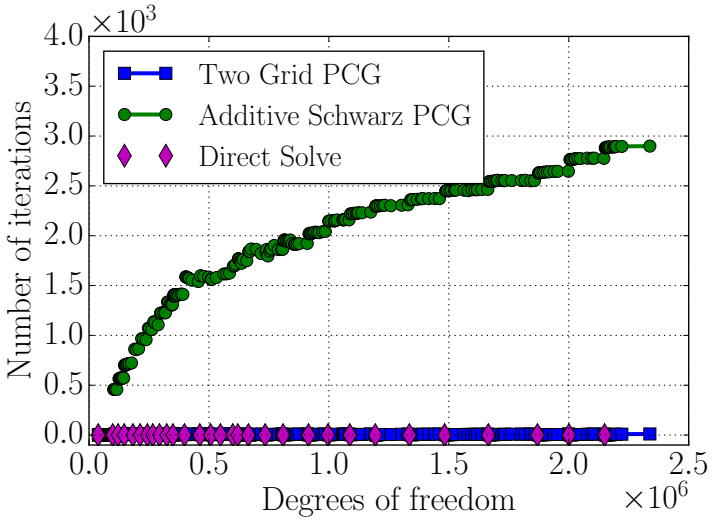


Figure 5.11: PCG solver for $\omega = 1500\pi$: iteration count vs skeleton dof

5.4 Computational cost

The total computational cost of the presented algorithm involves four major parts: a) the use of a direct solver on the coarse mesh, b) the construction of the macro grid c) the construction of the preconditioner and d) the CG iterations.

Direct solver on the coarse mesh. The direct solver is invoked for the solution on the coarse mesh. We use a multi-frontal solver, where its computational cost depends on the sparsity of the stiffness matrix which in turn depends on the mesh and the dimension. For uniform meshes, depending on reordering of the matrix, the cost is estimated to be $\mathcal{O}(N^{\frac{3}{2}})$ in $2D$ and $\mathcal{O}(N^2)$ in $3D$ [88], where N is the size of the system. For adaptive hp meshes it is very hard to come up with a good estimate since the refinements depend on the problem. For the particular example of this document, the simulation of a beam, we observed lower complexity of than $\mathcal{O}(N^{\frac{3}{2}})$ but higher than $\mathcal{O}(N)$. The purpose of the presented iterative algorithm is to avoid using the direct solver for several refinements. The goal is to replace the direct solver

with an iterative one of linear complexity, for most of the adaptive solves.

Construction of the macro grid. This step involves the elimination of the degrees of freedom of the fine mesh that are not on the skeleton of the coarse mesh. The procedure involves the solution of a local problem on each coarse grid element. The cost depends on the refinement pattern. For instance, in the standard two-grid setting, where the fine grid is obtained by a single uniform h-refinement of the coarse grid, the cost of the local problem is negligible. However, in the adaptive setting, there are cases where some coarse grid elements are refined several times, and some are not refined at all. This creates two major issues. First, the local problem corresponding to a coarse element which is refined several times is no longer of negligible size. Assuming that for its solution we use a sparse direct solver, then the cost is $\mathcal{O}(N_l^{\frac{3}{2}})$ in $2D$ and $\mathcal{O}(N_l^2)$ in $3D$, where N_l is the size of the system. Consequently, to keep the cost under control, then N_l has to be much smaller than the size of the global system. This can be achieved by reseting the coarse grid. In 3D computations, this problem is avoided by extending the two-grid technology to the multigrid setting. The multigrid algorithm is discussed in the next chapter. The second issue has to do with parallelization and work balancing. In the adaptive setting, workload is not the same in each coarse element, and therefore, unavoidably there is work imbalance among the possessors. While, it is not a major problem for shared memory implementations, it is significant for distributed memory implementations (MPI), and work balancing techniques have to be employed.

Construction of inter-grid and and smoother patches. The construction of the pre-conditioner involves a coarse grid correction and a smoothing process. For the coarse grid correction the construction of the prolongation and restriction operators is local, and the cost is negligible. Additionally, the coarse solve involves only back substitution since the Cholesky decomposition of the stiffness matrix of the coarse grid is already computed and stored. Therefore the cost is $\mathcal{O}(nz_c)$ where nz_c is the number of non-zero entries of the Cholesky factors. In fact if $nz_c = \mathcal{O}(N_c)$, where N_c is the size of the system in the coarse grid, then the cost

of back substitution is $\mathcal{O}(N_c)$. For the smoothing steps, a local construction of each vertex patch is required. This corresponds to the assembly and Cholesky factorization of individual blocks of the global stiffness matrix. These operations are local to each vertex patch, and therefore, when the size of each patch is much smaller than the size of the total system, the cost is negligible.

Smoothing and CG iterations. The overall cost of the solver is dominated by the global operations. The additive Schwarz smoothing is a global operation since the vertex patches overlap. The cost of a smoothing step is determined by the cost of a sparse matrix-vector multiplication. Therefore, the cost is linear with respect to the number of the non-zero entries of the smoothing matrix. We note that for a fixed order of approximation the number of non-zero entries is in fact $\mathcal{O}(N)$. Finally, there is an additional cost of the matrix-vector multiplications in the CG procedure. Again, the cost grows linearly with the size of the global stiffness matrix.

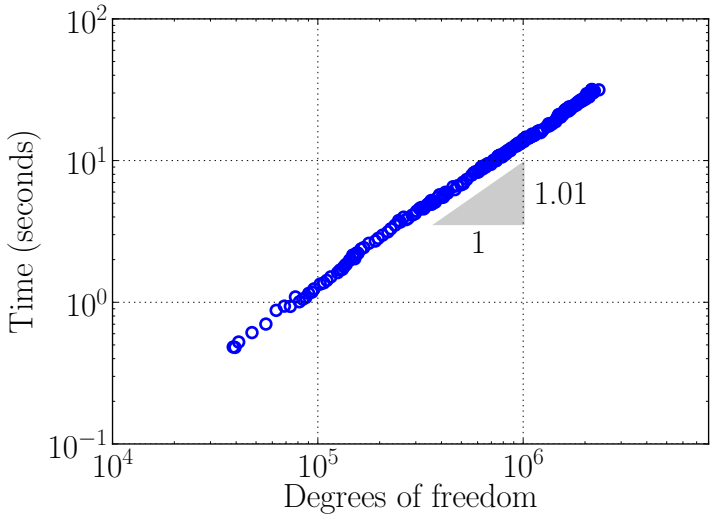


Figure 5.12: Time per iteration for the two-grid PCG. Throughout the adaptive process the cost of the solver remains linear with respect to the number of skeleton degrees of freedom

In Fig. 5.12 we show the cost of each iteration of the CG algorithm with respect to the size of the system for the cavity problem of Section 5.3.2. The time measurements are for a single core implementation. As it can be verified from the graph, the computational cost of each CG iteration increases linearly with respect to the size of the linear system. The total cost can be calculated by multiplying the cost of each iteration by the number of iterations. Therefore, as long as the number of iterations remains bounded, the overall computational cost of the solver grows linearly with the number of degrees of freedom.

Chapter 6

A 3D multigrid preconditioner

In this chapter we present an extension of the two-grid preconditioner, discussed in the previous chapter, to the multigrid setting. After giving a brief outline on the implementation, we provide actual time and memory measurements for a particular simulation, in the uniform refinement setting. We conclude the chapter by presenting timings for a parallel implementation for shared memory architectures.

6.1 Discussion on implementation

The major parts of the multigrid preconditioner are: a) construction of the macro-grids, b) construction of inter-grid operators, and c) assembly and solution of Schwarz local problems. All these parts of the construction are largely based on the two-grid preconditioner described in the previous chapter. For simplicity, here we discuss the three-grid case. Consider a fine grid which is the grid where we seek the solution, an intermediate grid and a coarse grid. In this scenario, we assume that the intermediate grid and the fine grid are constructed by an adaptive *hp*-refinement of the coarse and the intermediate grid respectively.

6.1.1 Construction of macro-grids

In the multigrid setting, each macro-grid is constructed by statically condensing the degrees of freedom which do not live on the mesh skeleton of the previous grid. This is essentially the computation of the Schur complement with respect to the degrees of freedom that live on the skeleton of the grid one coarse level below. In our three-grid scenario, one macro-grid is constructed by eliminating the fine-grid degrees of freedom that are not on the skeleton of the intermediate grid. Likewise, another macro-grid is constructed from the

intermediate and the coarse grid. This procedure is similar to the construction of the macro-grid in the two-grid case (see Section 5.1). However, there is a compelling advantage; the cost of this local elimination is significantly reduced. Unlike the two-grid setting, where the local Schur complement problem would correspond to arbitrary many refinements of a coarse-grid element, in the multigrid setting the size corresponds to only one refinement. The construction is illustrated in Figure 6.1. In the next section, we will demonstrate that the cost of this construction grows linearly with the number of unknowns.

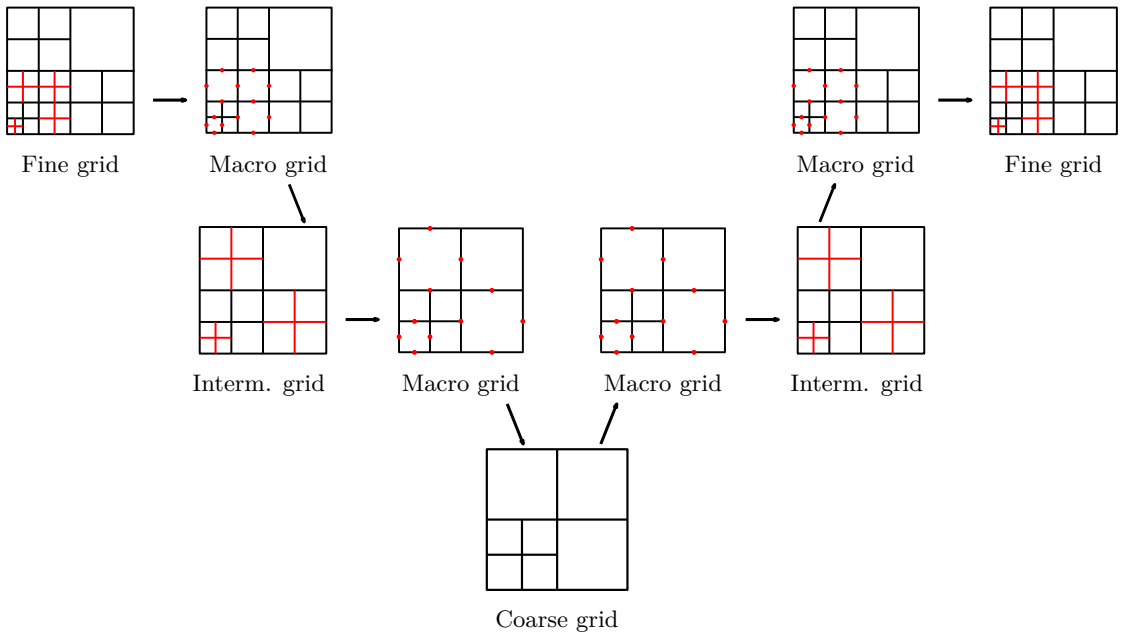


Figure 6.1: Multigrid v-cycle schematic. For demonstration purposes the schematic is in 2D. In 3D it is fully analogous

6.1.2 Inter-grid transfer operators

Recall that the DPG linear system involves only interface unknowns living on the mesh skeleton. As described in Section 5.1 this adds a complication when constructing the inter-grid transfer operators. In the two-grid setting we managed to overcome this complication by constructing a single macro-grid from the fine grid, and then applying the two-grid cycle between macro and coarse grid. Following this approach, in the multigrid setting, we have

two inter-grid operators, a Schur complement extension/restriction operator and a natural inclusion/restriction operator. The multigrid cycle is then as follows (see Figure 6.1). Given solution degrees of freedom on the coarse grid, we apply the natural inclusion operator to obtain solution degrees of freedom on the macro-grid. After smoothing on the macro-grid, the Schur complement extension operator is employed to compute the solution degrees of freedom to the intermediate grid. With the same procedure, we compute the solution degrees of freedom on the fine-grid. The restriction operators that act on the residual are defined to be the transposes of the inclusion operators. Note that the Schur complement inclusion/restriction operators can be computed and stored during the construction of the macro-grids.

6.1.3 Local Schwarz problems

In the multigrid setting described above, we choose to perform the smoothing operations on the macro-grids. A smoother patch is defined to be the support of a vertex basis function of the grid one coarse level below. Then, a local assembly within each patch is performed, and a Cholesky decomposition is invoked. The decompositions are computed once and stored. They are then used in each smoothing step within each multigrid cycle. Note that the size of each local problem remains fixed and therefore the total cost of the construction of the smoother is of linear complexity. This will also be demonstrated in the next section.

6.2 Computational complexity

In this section we study the computational complexity of the multigrid technology by considering a particular example in three space dimensions. We demonstrate that the overall computational cost grows linearly with respect to the size of the linear system. Comparison with a state of the art multi-frontal solver is also presented. Before we proceed to the example it is essential that we provide some additional details regarding the numerical implementation of our experiments within our in-house 3D code (hp3D).

6.2.1 Additional implementation details

Static condensation of interior degrees of freedom. Static condensation is a very common procedure in finite element methods. Especially, in high order methods the size of the global system is dramatically reduced by eliminating the degrees of freedom associated with the element interior. In the ultraweak DPG formulation all the L^2 degrees of freedom have no global conformity requirement and therefore they are eliminated in an element-wise fashion. The resulting linear system then involves only the trace unknowns defined on the mesh skeleton. The Schur complements can be either stored or recomputed when needed. After the interface problem is solved, either with a direct or an iterative solver, using local back substitutions the interior solution degrees of freedom are recovered.

Fast integration. Computing the local stiffness matrices is one of the most computationally intensive parts in high order finite element methods. For instance, for a hexahedral element, the standard Gaussian integration has a complexity of $\mathcal{O}(p^9)$, where p is the order of approximation in each space direction. In DPG simulations, the need for an enriched test space (see Remark 2.2) adds significant computational work on the element level. Therefore, employing fast integration techniques is essential, in order to keep the cost under control. For all the 3D simulations in this work, fast integration techniques based on tensorization are used. For the hexahedral meshes that we use in our simulations, the integration cost is reduced from $\mathcal{O}(p^9)$ to $\mathcal{O}(p^7)$. We refer the reader to the work of Jaime Mora [95] and references therein for a detailed discussion on implementation and results.

Multigrid: global matrix-free implementation. In any finite element code, the global linear system can be given either in a local or in an assembled form. By local form we mean that the local stiffness matrices and load vectors are computed but a global stiffness matrix and load vector are never assembled. Instead, only the local to global connectivity maps are computed and used only when needed. Even if this approach might have higher memory requirements (information on shared degrees of freedom is computed and stored for all neighboring elements),

for our multigrid preconditioner it is preferable. We choose the unassembled approach, mainly for one reason. This way, the construction of all components of the preconditioner remains local, and that gives an explicit control on parallelization. Additionally, future work involves extending this construction in distributed memory environments using MPI, aiming problems where assembly of a global stiffness matrix on one compute node would not be feasible.

Multi-frontal solver: assembly in CSR format. When a direct solver is needed we use the multi-frontal solver PARDISO which is available within the MKL library. Interfacing with this solver requires the global stiffness matrix in assembled form, given in Compress Sparse Row (CSR) format. In our code, we obtain this format by first assembling in parallel the global stiffness matrix in a coordinate (COO) format and then by using a parallel version of quick sort (complexity of $\mathcal{O}(NlogN)$), the COO format is converted to the CSR format. We note that for large implementations the sorting can take a non negligible percentage of time of the solving stage (approximately 5%). For the results on timing presented in the next section the sorting time is not included. For a detailed discussion on different storage formats of sparse matrices we refer the reader to [110, Ch. 3].

6.2.2 Set up

We solve the linear acoustics problem in the unit cube using the DPG ultraweak formulation. The exact solution for the pressure is a plane wave propagating from the origin in the direction $k = (1, 1, 1)$, i.e, $p_{\text{exact}} = e^{-i\omega(x+y+z)}$, and it's shown in Figure 6.2. The simulation is driven by an impedance condition on the entire boundary, where the impedance data is lifted from the exact solution, and the frequency is $\omega = 3\pi$. The ultraweak formulation is given by:

$$\begin{cases} u \in (L^2(\Omega))^d,\ p \in L^2(\Omega), \hat{u}_n \in H^{-1/2}(\Gamma_h),\ \hat{p} \in H^{1/2}(\Gamma_h) \\ \hat{p} - \hat{u}_n = g,\ \text{on } \partial\Omega \\ (i\omega u, v) - (p, \text{div}_h v) + \langle \hat{p}, v \cdot n \rangle_{\Gamma_h} = 0, \quad v \in H(\text{div}, \Omega_h) \\ (i\omega p, q) - (u, \nabla_h q) + \langle \hat{u}_n, q \rangle_{\Gamma_h} \quad = 0, \quad q \in H^1(\Omega_h) \end{cases}$$

where $\Omega = (0, 1)^3$ and Γ_h is the mesh skeleton.

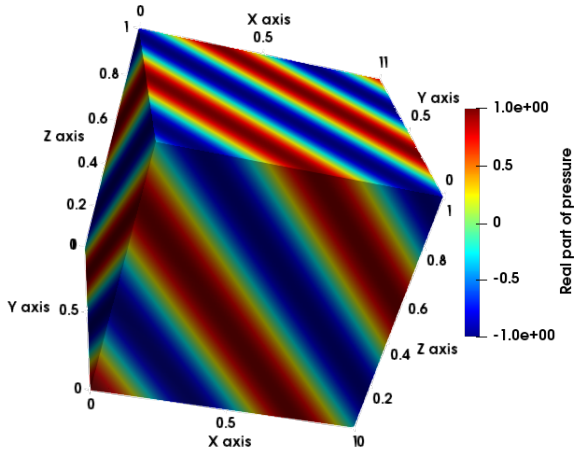


Figure 6.2: Plane wave exact solution

6.2.3 Direct solver

First we run the simulation using a direct-solver and verify that the theoretical rate is recovered. We start with a uniform mesh of 8 cubic hexahedral elements, and perform 4 successive uniform $h-$refinements. Since the solution is smooth, the expected rate of convergence is h^p, or $N^{-p/d}$, where N is the number of unknowns, p is the polynomial order corresponding to the exact sequence (see Section 2.4.2.2) and d is the dimension. The convergence is shown in Figure 6.3.

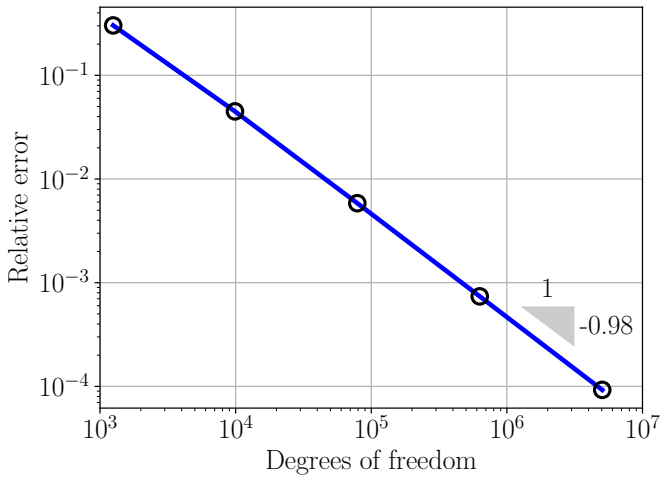


Figure 6.3: Error vs dof using a direct solver: for $p = 3$ the optimal convergence rate is -1

Usually, in 3D computations the linear solve stage is the part with the highest computational intensity. The pie chart below (see Figure 6.4) shows the time distribution of the numerical simulation of our example for the last mesh. The mesh consists of 32768 cubic elements with total number of degrees of freedom of approximately five million. As we can see, almost 90% of the total simulation time is spent on the linear solver. Recall that in the adaptive refinement setting, the problem has to be solved multiple times, and employing a direct solver at every step is obviously not optimal. This is the motivation behind the construction of our multigrid preconditioner.

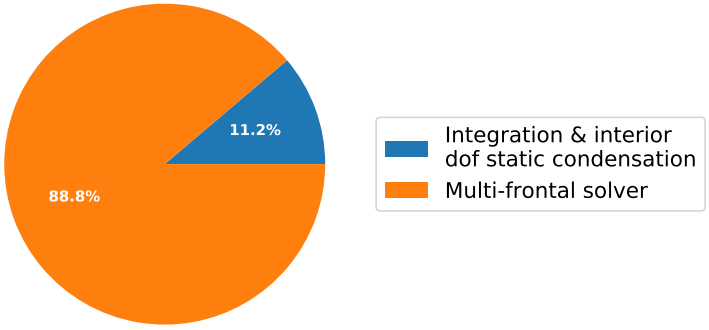


Figure 6.4: Distribution of overall computational cost when a direct solve is used

6.2.4 CG solver preconditioned with multigrid

We present now computational complexity measurements of the construction of the multigrid preconditioner, and we demonstrate that both time and memory grows linearly with respect to the size of the linear system. The set up is as follows. We define our coarse grid to be the initial mesh of the simulation, consisting of eight cubic elements. This mesh provides a solution with relative error of approximately 30%. Then for each refinement we run the CG solver, preconditioned with the multigrid preconditioner. The multigrid preconditioner, involves smoothing at all the intermediate grids and an exact solve using a direct solver at the coarse level. As described in the previous section the construction of the iterative solver can

be broken down to four components.

- Prolongation operator: computation of the coefficients for the natural inclusion operator.
- Macro-grid: construction of the macro grid and Schur complement extension operators
- Schwarz patches: assembly and factorization of patch stiffness matrices corresponding to a Schwarz smoother patch.
- CG iterations: the actual solving step, including the application of the preconditioner (smoothing and coarse grid solves).

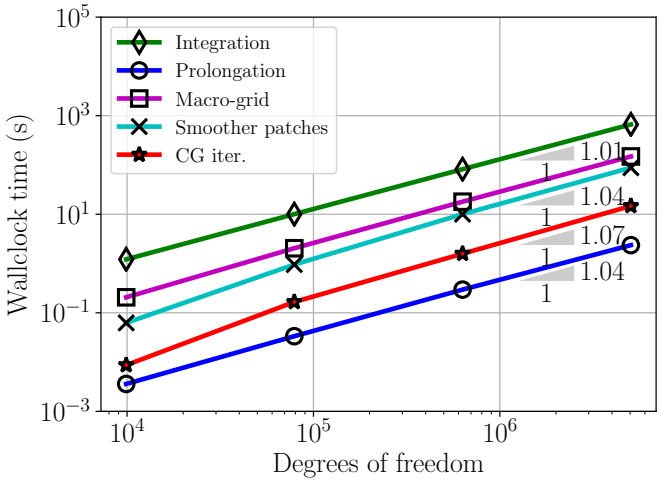
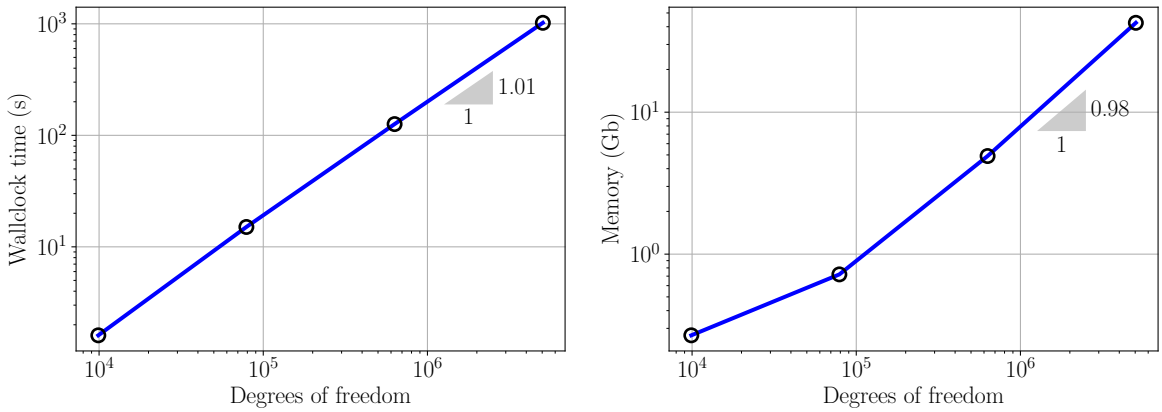


Figure 6.5: Timing measurements for all the different components of the numerical simulation (serial)

In Figure 6.5 we show a timing breakdown for all four components of the solver along with the construction of the local stiffness matrices (integration). It is interesting to see that now the most expensive part is the numerical integration. Additionally, we can conclude that all the different parts scale linearly with respect to the number of unknowns. We therefore expect that the cost of the whole simulation will also grow linearly with the size of the system too. Finally, the iterative solver is based on matrix vector multiplications and so only the local stiffness and load vectors are needed to be stored (or a sparse representation of the global stiffness matrix). Consequently, the required memory is also expected to grow linearly with the size of the problem. These conclusions are shown in Figure 6.6.



(a) Total time in seconds vs dof

(b) Total memory in GB vs dof

Figure 6.6: Overall time and memory needed by the simulation using the multigrid preconditioner. Linear dependence on the number of degrees of freedom is observed for both the computation time and the memory requirements.

We conclude this chapter by presenting three additional plots. The first one Figure 6.7 shows the time distribution for the whole simulation. As desired the solve time now is actually less than the integration time. The second plot reveals the well known result for multi-frontal direct solvers in 3D, that they scale quadratically with respect to the size of the system. In this plot we show only the timings involving only the solution stage, i.e, the integration times are excluded.

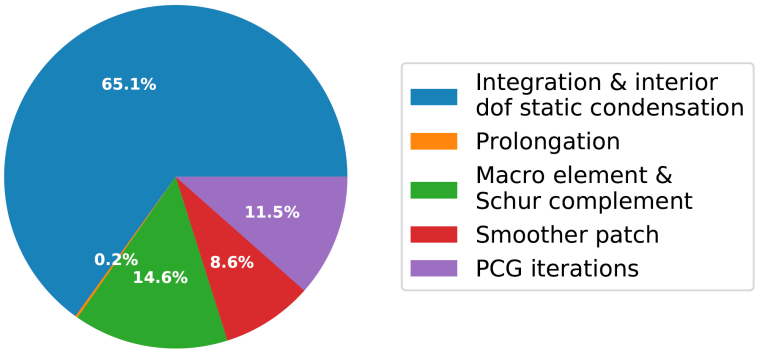


Figure 6.7: Distribution of time for the whole simulation when using the multigrid preconditioner

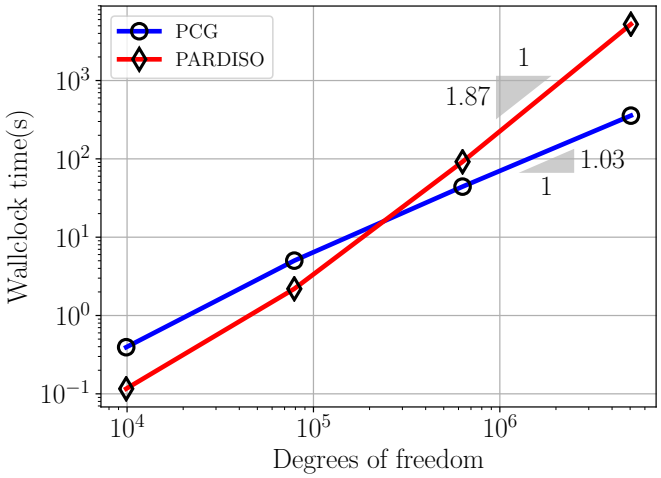


Figure 6.8: PCG vs PARDISO timing measurements. As the theoretical estimates suggest the PCG preconditioner is of linear complexity. On the contrary the multi-frontal solver asymptotically reaches quadratic complexity.

Finally, in Figure 6.9 we show plot the error against the overall cost when using the multigrid preconditioner for the solution stage. Since that solver's computation complexity is linear and since the theoretical convergence rate for this particular configuration is -1, the expected slope of the line is also -1. This result demonstrates that no accuracy is lost by reducing the overall cost of the simulation.

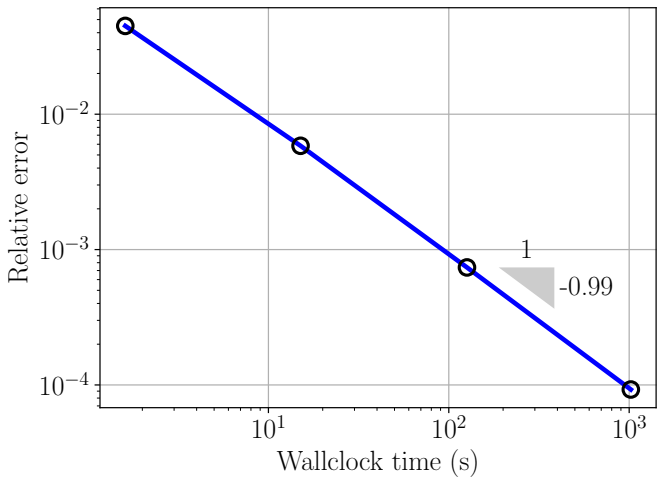


Figure 6.9: Error vs time using the PCG solver

6.3 Parallel implementation

We conclude this chapter by presenting results on a particular parallel implementation of the multigrid preconditioner. The parallel version of the preconditioner was implemented for shared memory architectures using OpenMP. The simulation was run on a single NUMA node, consisting of two sockets with 12 cores each. We emphasize that the purpose of this parallel study is by no means to show perfect parallelization scalings, but rather to investigate the parallel efficiency of the various components of our construction. Upcoming projects within our research group will involve the development of an MPI version of our code and therefore this investigation is of significant importance.

We run the same experiment outlined in Section 6.2.2 on the finest mesh of 32768 cubic hexahedral elements. The size of the system is kept fixed and we measure the wall-clock times of the simulation for 1, 2, 4, 8 and 16 processors. In Figures 6.10 and 6.11 we present strong scaling speedups for the different components of the iterative solver and the overall simulation respectively. Observe that the components of the simulation that do not need any communication show near optimal speedups. We can argue that the slight diversion of the speedups from the optimal one is due to our computer architecture. That is, in order to utilize 16 cores both sockets have to be used, and therefore it is possible that a core from one socket would have to access memory from the other socket.

The most interesting observation is with regards to the speedup of the CG iterations. The cost is dictated by the action of the preconditioner, which includes the action of the smoother and the coarse grid correction. The smoother, by construction, involves communicating information among patches. Every local Schwarz problem can be solved locally in a parallel fashion but a global correction to the solution has to be computed by assembling the local solutions from each patch. Since the patches are not disjoint, there is unavoidable communication. In practice, this assembly procedure is implemented using the OpenMP *reduction clause*, for which it can be shown that the expected speedup is less optimal than linear with respect to the number of processors.

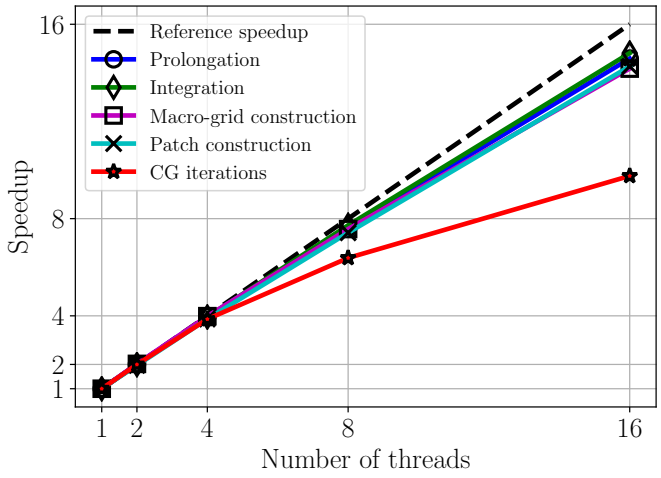


Figure 6.10: Time speedup on a shared memory architecture for each component of the numerical simulation

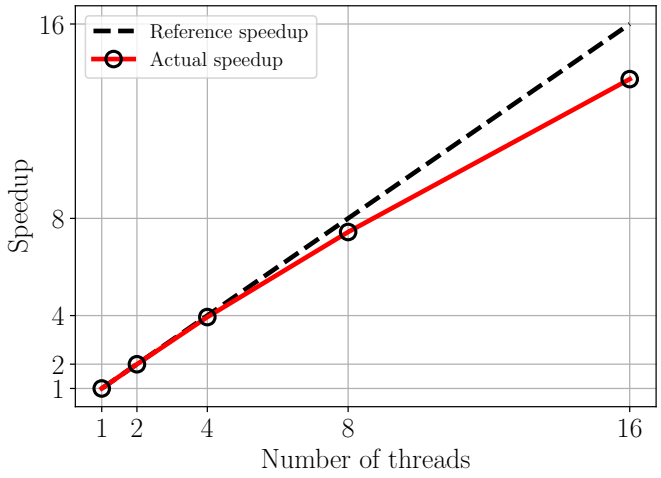


Figure 6.11: Time speedup on a shared memory architecture for the complete numerical simulation

Chapter 7

Numerical results in 3D

This chapter is devoted to numerical results for relatively large simulations, using the Conjugate gradient solver preconditioned with the multigrid technology. Our interest is in computationally challenging acoustics and electromagnetic problems. We present various numerical experiments in acoustics and Maxwell equations in both uniform and adaptive refinement setting.

7.1 Time harmonic Maxwell equations

We first consider the time harmonic form of the Maxwell equations given by

$$\begin{cases} i\omega\mu H + \nabla \times E = 0 \\ -i\omega\epsilon E + \nabla \times H = J \end{cases}$$

where E is the electric field, H is the magnetic field, ω is the angular frequency, ϵ is the permittivity and μ and permeability of the material.

7.1.1 Comparison with standard multigrid methods

For our first Maxwell experiment we compare our DPG multigrid technology with the multigrid preconditioner for the standard Galerkin method described in [64]. Following the experiment in [64], we consider the computational domain $\Omega = (0,1)^3$ and a *perfect electric conductor* (PEC) material, i.e, the boundary condition is $n \times E = 0$ on $\partial\Omega$. For the discretization we use a uniform hexahedral mesh of the lowest order. Define the space $H_0^{-\frac{1}{2}}(\text{curl}, \Gamma_h)$ on the mesh skeleton as

$$H_0^{-\frac{1}{2}}(\text{curl}, \Gamma_h) := \{E \in H^{-\frac{1}{2}}(\text{curl}, \Gamma_h) : n \times E = 0 \text{ on } \partial\Omega\}$$

where $H^{-\frac{1}{2}}(\text{curl}, \Gamma_h)$ is as in (2.17). Then the ultraweak DPG formulation is

$$\begin{cases} E, H \in (L^2(\Omega))^3, \\ \hat{E} \in H_0^{-\frac{1}{2}}(\text{curl}, \Gamma_h), \ \hat{H} \in H^{-\frac{1}{2}}(\text{curl}, \Gamma_h) \\ i\omega\mu(H, F) + (E, \nabla_h \times F) + \langle n \times \hat{E}, F\rangle_{\Gamma_h} = 0, \qquad F \in H(\text{curl}, \Omega_h), \\ -i\omega\epsilon(E, G) + (H, \nabla_h \times G) + \langle n \times \hat{H}, G\rangle_{\Gamma_h} = (J, G), \quad G \in H(\text{curl}, \Omega_h) \end{cases}$$

where the letter h denotes element-wise operations. The simulation is driven by the right hand side which is chosen such that the exact solution for the electric field is given by the finite element lift of the function

$$E_{\text{ex}}(x, y, z) = [y(1-y)z(1-z), yx(1-x)z(1-z), x(1-x)y(1-y)]$$

We choose $\epsilon = \mu = 1$. The initial guess for the CG solver is zero and the iterations are terminated when the norm of the residual is reduced by a factor of 10^{-6}. In Tables 7.1a and 7.2b we present the iteration count of the CG solver when preconditioned with our multigrid technology for $\omega = 1$ and 10. Likewise, Tables 7.1b and 7.2b, retrieved from [64], show the iteration count of the GMRES solver preconditioned with the standard multigrid technology. We note that the smoother used in [64], is of multiplicative type (block Gauss-Seidel). In the tables, h and H denote the fine and the coarse grid discretization size respectively. For the DPG multigrid implementation, On each level we perform one pre- and one post- smoothing step, so that we keep the symmetry. At the coarsest level the problem is solved exactly with a direct solver.

h \H	1/2	1/4	1/8	1/16
1/4	7			
1/8	7	7		
1/16	7	7	7	
1/32	6	7	6	7
1/64	6	6	6	6

(a) Iteration count of CG solver preconditioned with DPG multigrid

h \H	1/2	1/4	1/8	1/16
1/4	6			
1/8	7	7		
1/16	9	10	8	
1/32	10	10	9	7
1/64	11	11	9	8

(b) Iteration count for GMRES preconditioned with multigrid for standard Galerkin

Table 7.1: Iteration count for $\omega = 1$. Observe the uniform convergence with respect to h and H.

h \H	1/2	1/4	1/8	1/16
1/4	9			
1/8	11	10		
1/16	14	12	9	
1/32	14	14	12	10
1/64	15	15	12	12

(a) Iteration count of CG solver preconditioned with multigrid for the DPG method

h \H	1/2	1/4	1/8	1/16
1/4	3*			
1/8	2*	37		
1/16	3*	48	18	
1/32	2*	78*	22	16
1/64	2*	78*	21	17

(b) Iteration count for GMRES preconditioned with multigrid for standard Galerkin

Table 7.2: Iteration count for $\omega = 10$. The number of iterations for the DPG method grows mildly with the frequency but always converges to the true solution. Uniform convergence is achieved when a fine enough coarse grid is used. On the contrary the GMRES method fails to deliver reliable solutions when the coarse grid is in the pre-asymptotic region.

The purpose of this comparison is not to compare number to number the iteration count for each multigrid technology, but rather to observe the general convergence trend of our preconditioner compared to the preconditioner for the standard Galerkin method. The entries n^* in the Table 7.2b denote that even though the GMRES solver did converge, the final iterate differed from the true solution by more than 10^{-3} (measured in the appropriate norm). This is a well known flaw of the GMRES solver, i.e, the residual of the GMRES algorithm might be small enough and the stopping criterion is met, but the output solution is far from the true solution. This happens when the coarse grid is not fine enough for the corresponding frequency and stability is lost. On the contrary, this undesirable convergence behavior is not happening for the Conjugate Gradient solver. The discrete pre-asymptotic stability of the DPG method, along with the theory of self-adjoint preconditioners, ensure that the CG solver always converges to the true solution (see Section 4.2.1). However, the convergence does depended on how "good" the coarse grid is with respect to the frequency. A similar dependence on the frequency is observed in our 2D simulations (see Section 5.2.2). For both preconditioners, uniform convergence with respect to the frequency is recovered when the coarse grid is fine enough.

7.1.2 Fichera "oven" problem

For our second Maxwell experiment we solve the Fichera "oven" problem which was first presented in [19]. The adaptive nature of the DPG method makes it suitable for this problem because the solution is expected to be singular. The set up is as follows. For the construction of the domain we start with the cube $(0,2)^3$ which is uniformly refined into eight cubes and then one is removed creating the Fichera corner. Then, an infinite waveguide is attached at the top and it's truncated at a unit distance from the Fichera corner (see Figure 7.1). We choose $\epsilon = \mu = 1$ and $\omega = 5$. The simulation is driven by a non-homogeneous electric boundary condition on the waveguide and a homogeneous electric boundary condition elsewhere. That is,

$$n \times E = \begin{cases} n \times E_d & \text{across the waveguide section} \\ 0 & \text{elsewhere} \end{cases}$$

where $E_d = (\sin \pi x_2, 0, 0)$ is the first propagating mode.

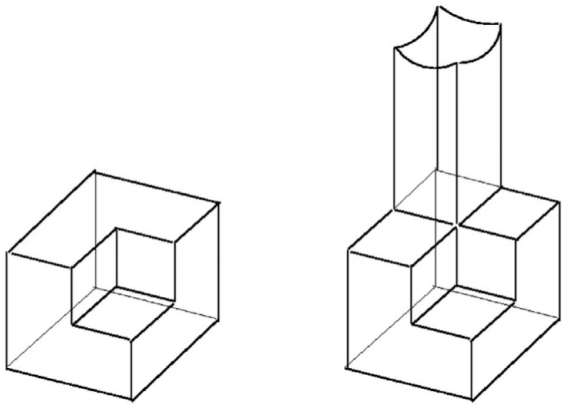
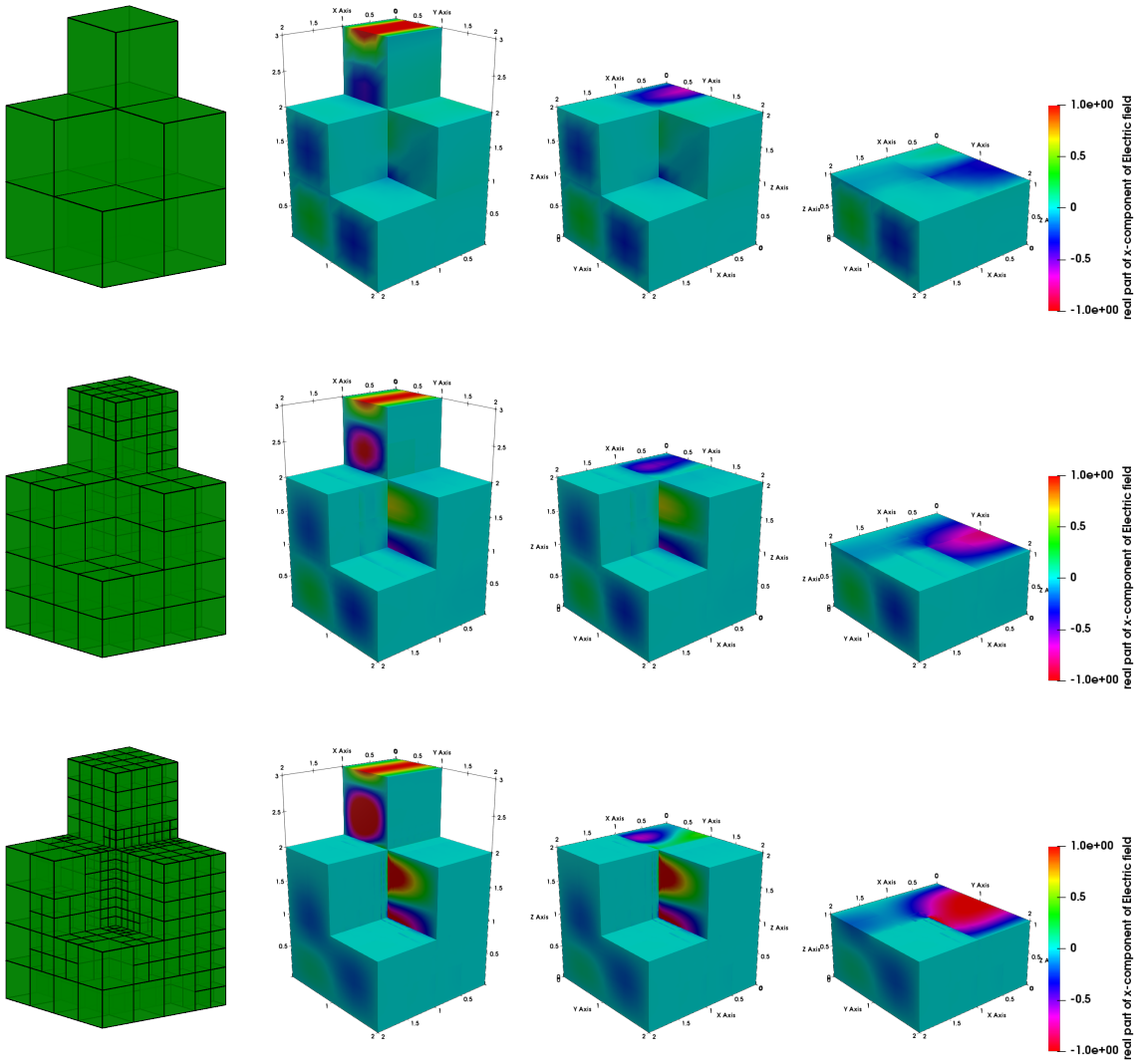
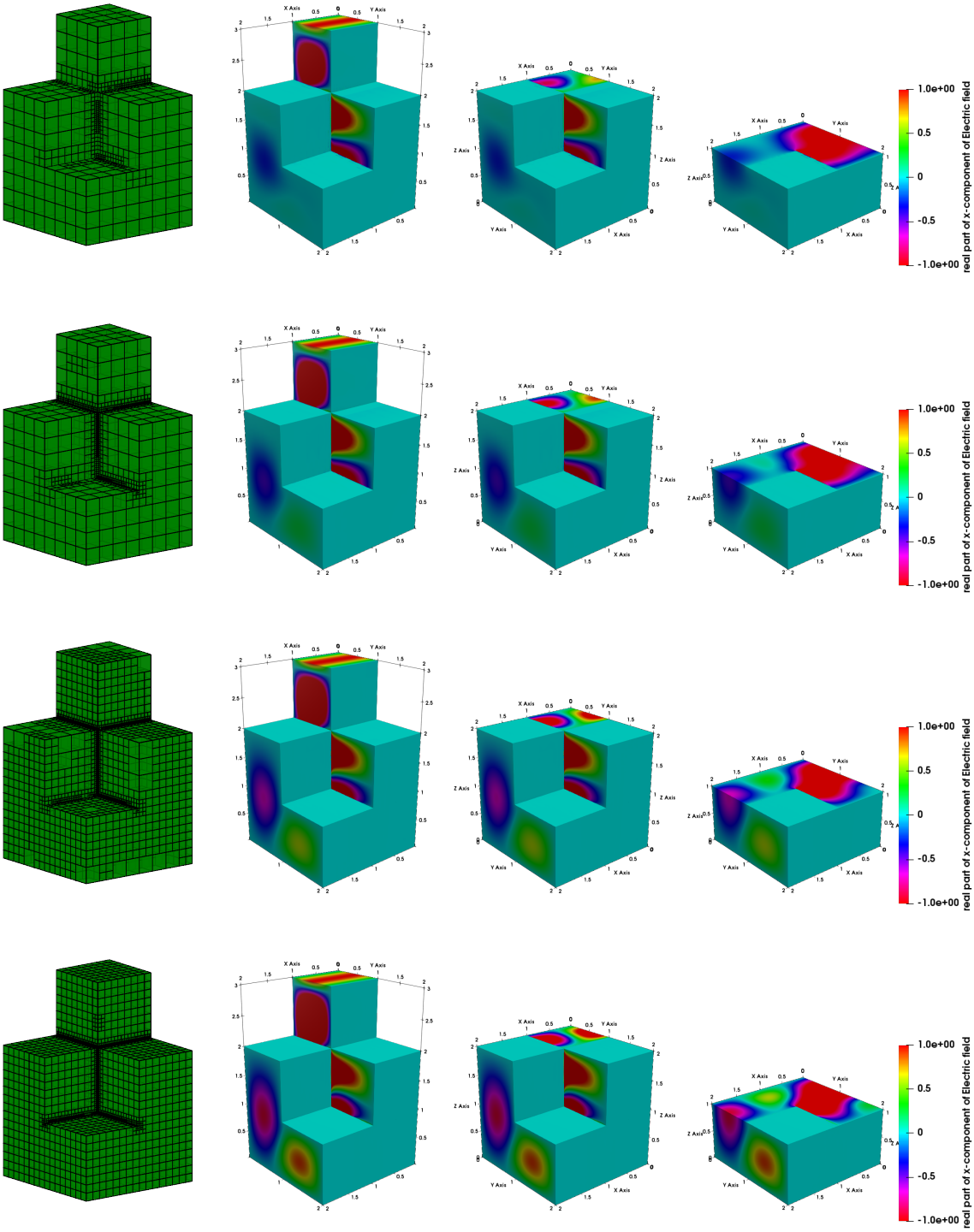


Figure 7.1: Fichera corner with a truncated infinite waveguide attached at the top (retrieved, January 10, 2019 from [19])

We start the simulation with a mesh of eight cubes with a uniform order of approximation $p = 3$ and perform successive $h-$adaptive refinements. The adaptive refinements are driven by the built-in DPG error indicator, and terminated when the norm of the residual decreases by one order of magnitude. Note that an exact solution for this problem is not known.

The multigrid preconditioner setting is as follows. Starting from the finest mesh, several coarser adaptive meshes, which belong to the history of refinements, are selected in the multigrid cycle. An exact solve is performed at a coarse grid, where the size of the system is significantly smaller than the fine grid system and small enough to be solved efficiently with a direct solver. The selection is made at run time and depends on the computer architecture and the current memory available.




X Axis
Y Axis
Z Axis
1.0e+00
0.5
0
-0.5
-1.0e+00
real part of x-component of Electric field

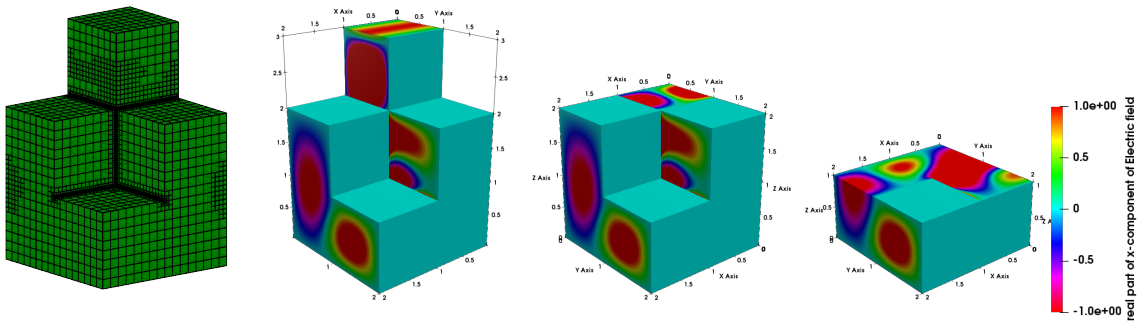
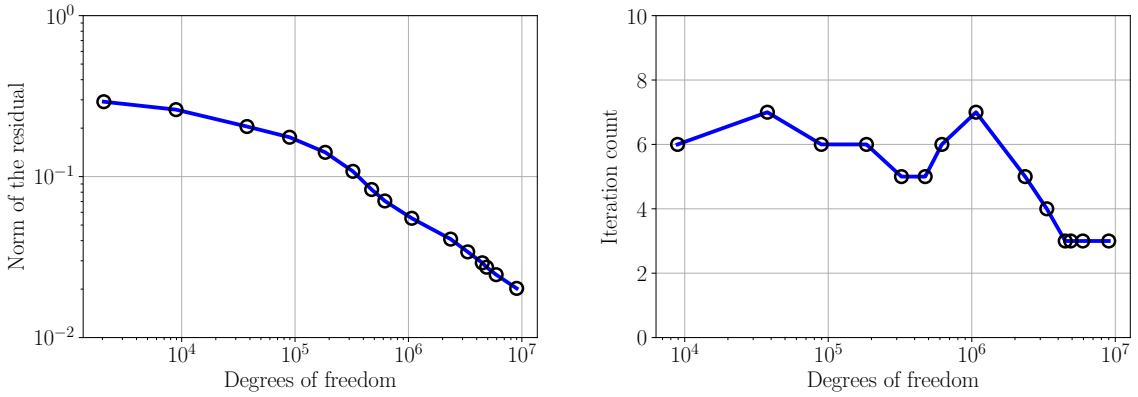


Figure 7.2: Evolution of the mesh and the numerical solution for the real part of the x-component of the electric field. These results are for the meshes 1,3,5,7,9,11,13 and 15.

In Figure 7.2 we show the evolution of the mesh along with the numerical solution for the real part of the x-component of the electric field. As we can see, the ultraweak DPG method, captures the singularities at the reentrant corner and edges very fast. Observe how the solution changes qualitatively and quantitatively with the resolution of the singularities.



(a) Residual convergence

(b) Iteration count for the preconditioned CG solver

Figure 7.3: Fichera problem: convergence of the residual (left) and the CG solver(right)

The convergence of the residual is displayed in Figure 7.3a. Since there is no known exact solution to this problem, the only way to quantify the convergence is through the DPG residual. Clearly the norm of the residual converges to zero as we proceed with refinements.

In Figure 7.3b we illustrate the convergence behavior of the CG solver preconditioned with multigrid. For this simulation, at most five levels were used in the multigrid cycles. At all intermediate multigrid levels, we perform 10 smoothing steps with a relaxation parameter selected according to the finite overlap property (here $\theta = 0.2$). The CG iterations are initiated with a zero initial guess and they are terminated when the norm of the discrete residual is reduced below 10^{-6}. As we can observe, the solver shows uniform convergence with respect to the discretization size throughout the adaptive refinement process. We emphasize that the problem was too large to be solved with a direct solver.

7.1.3 Gaussian beam in free space

Our last Maxwell example involves the simulation of a high-frequency Gaussian beam scattering by a rigid cube. We consider the domain $(0,1)^3$ where a cube of side length $a = 1/7$ centered at $(0.5, 0.5, 0.5)$ is removed ($\Omega = (0,1)^3 \backslash (\frac{3}{7}, \frac{4}{7})^3$, see Figure 7.4). The simulation is driven by an impedance boundary condition on the outer cube and homogeneous electric boundary condition on the inner cube. The impedance data around the origin correspond to a high frequency Gaussian beam propagating inside the domain at a specific angle. Away from the source the impedance data smoothly decay to zero in order to simulate absorbing boundary conditions.

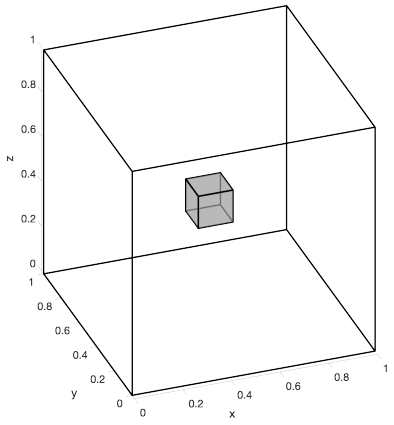


Figure 7.4: Computational domain

The DPG ultraweak formulation for this setting is given by:

$$\begin{cases} E, H \in (L^2(\Omega))^3,\ \hat{E} \in H^{-\frac{1}{2}}_{\Gamma_2}(\mathrm{curl}, \Gamma_h),\ \hat{H} \in H^{-\frac{1}{2}}(\mathrm{curl}, \Gamma_h) \\ n \times (n \times \hat{E}) - n \times \hat{H} = \hat{g}, \quad \text{on } \Gamma_1, \\ i\omega\mu(H, F) + (E, \nabla_h \times F) + \langle n \times \hat{E}, F \rangle_{\Gamma_h} = 0, \quad F \in H(\mathrm{curl}, \Omega_h), \\ -i\omega\epsilon(E, G) + (H, \nabla_h \times G) + \langle n \times \hat{H}, G \rangle_{\Gamma_h} = 0, \quad G \in H(\mathrm{curl}, \Omega_h) \end{cases}$$

where, Γ_1, Γ_2 are the boundaries of the outer and the inner cube respectively and

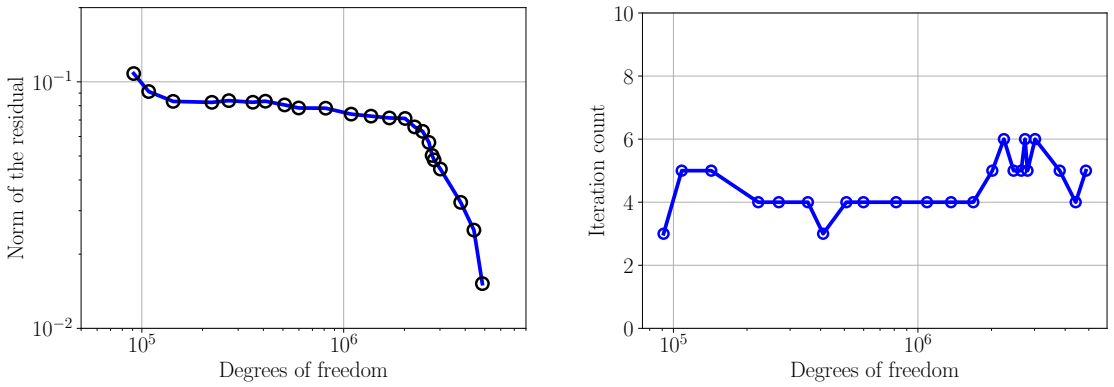
$$H^{-\frac{1}{2}}_{\Gamma_2}(\mathrm{curl}, \Gamma_h) := \{\hat{E} \in H^{-\frac{1}{2}}(\mathrm{curl}, \Gamma_h) : n \times \hat{E} = 0 \text{ on } \Gamma_2\}.$$

We start the simulation with a uniform mesh of 342 cubes of order $p = 3$ and we initiate adaptive *hp*-refinements. We follow an ad-hoc refinement strategy; an element marked for refinement is h-refined unless its size is less than half a wavelength, in which case it's p-refined (two elements per wavelength). In order to resolve the anticipated singularities on the scatterer, the elements adjacent to its corners and edges are marked and forced to be h-refined when needed.

Finally, when an adaptive p-refinement is performed, we follow the following rule (minimum rule). When an element is marked for a p-refinement we first find the neighboring elements with respect to its faces. The order of a neighboring element is then increased by one if it is less than the intended order of the marked element. The last step is to assign orders to edges and faces. The order of a face is defined to be the minimum of the orders of its neighboring elements. After all faces are assigned their order, the edge orders take the value of the minimum of the orders of the faces they belong to. With this rather complicated rule, we ensure that a p-unrefinement is not possible (the sequence of meshes remains nested), and the constrained approximation technology for handling hanging nodes [37] is well defined. We mention that a maximum rule was also tested successfully. However, in order to maintain stability the enriched order of the test space had to be increased, and this often led to significantly increased element computation times.

We run the simulation for $\epsilon = \mu = 1$ and $\omega = 50\pi$. This frequency corresponds to approximately 40 wavelengths inside the computational domain. The multigrid setting is the

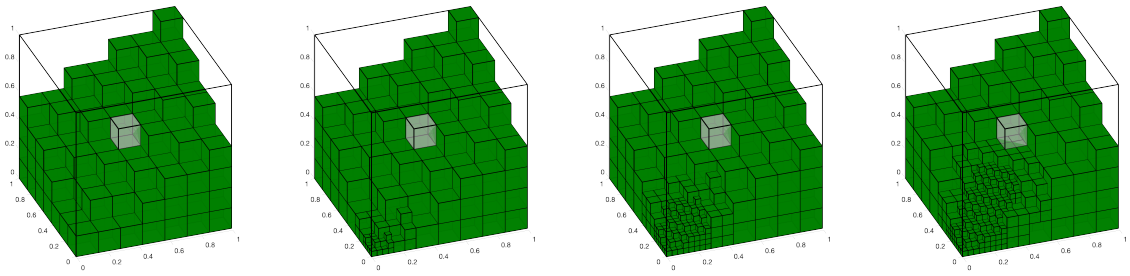
same an in the previous example. For this experiment, at most eight multigrid levels were used. Adaptivity is guided by the norm of the residual which is shown to be driven to zero in Figure 7.5a. The iteration count of the CG solver preconditioned with our multigrid technology is presented in Figure 7.5b. Note that the number of iterations of the CG solver remains under control throughout the adaptive refinement process. Lastly, the evolution of the mesh and the numerical solution of the real part of E_x are shown in Figures 7.6 and 7.7 respectively. Note that in these figures we show only the part of the domain below the plane defined by the point $(0.5, 0.5, 0.5)$ and the normal vector $(-0.5, -0.5, 1)$.



(a) Residual convergence

(b) Preconditioned CG iteration count

Figure 7.5: Convergence of the residual and the preconditioned CG solver



Refinement 1 Refinement 3 Refinement 5 Refinement 7

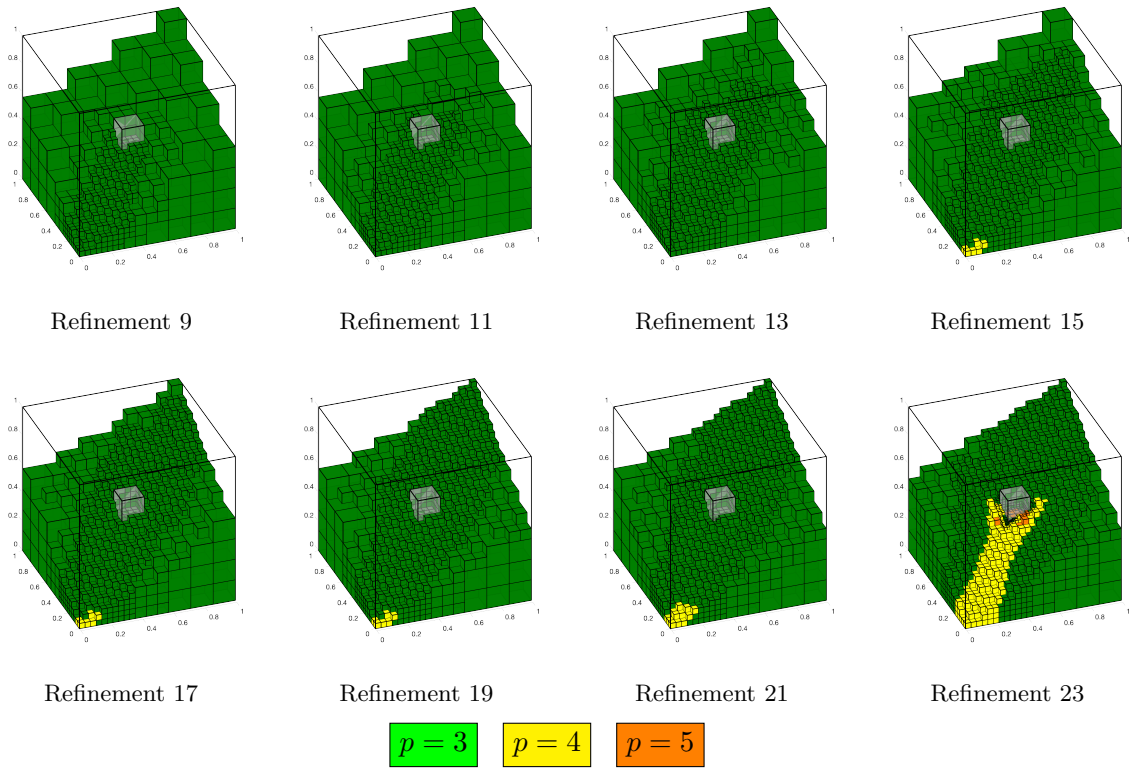
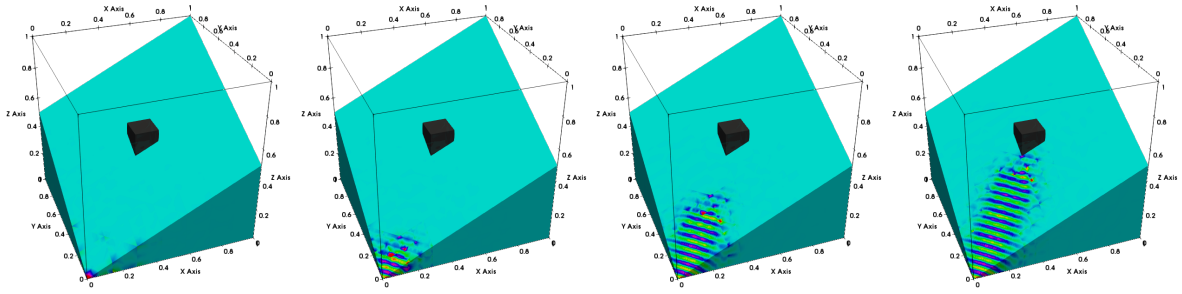


Figure 7.6: Evolution of the hp-adaptive meshes.



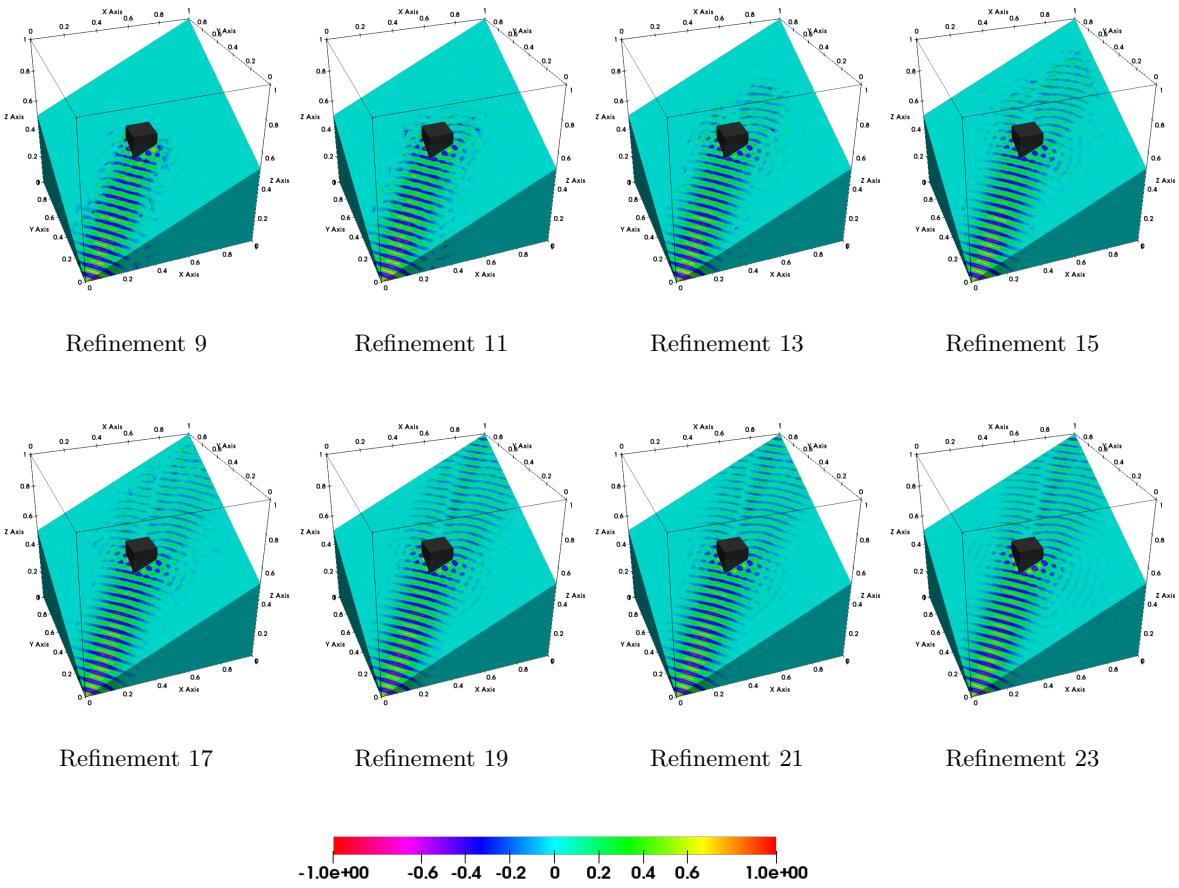


Figure 7.7: Real part of the numerical solution of the x-component of the electric field. Notice how the DPG adaptive technology refines only in regions of the domain where there is wave activity.

7.2 Linear acoustics equations

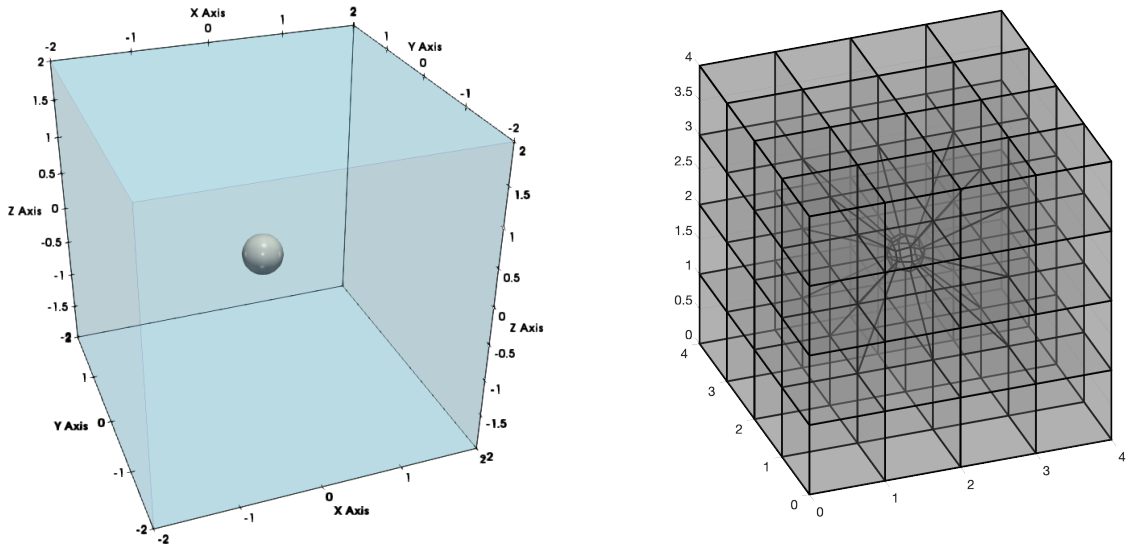
We continue with three dimensional examples for the linear acoustics problem. Recall the time harmonic form of the acoustics equations,

$$\begin{cases} i\omega p + \operatorname{div} u = f, \\ i\omega u + \nabla p = 0, \end{cases}$$

where ω is the angular frequency, p is the pressure and u is the velocity.

7.2.1 Plane wave scattering from a sphere

For our first example we consider the simulation of the scattered wave when a plane wave hits a rigid sphere (see Figure 7.8a). The domain is meshed[1] by hexahedra as shown in Figure 7.8b and the spherical surfaces are approximated using transfinite interpolation [37]. The incident plane wave is traveling in the direction $(1, 1, 1)$ with frequency $\omega = 35\pi$. That corresponds to approximately 30 wavelengths inside the domain.



(a) Domain

(b) Hexahedral mesh

Figure 7.8: Computational domain with a spherical scatterer and initial mesh

The simulation is driven by hard boundary conditions on the spherical boundary (Γ_{sph}),

$$u \cdot n = g = -u_{\text{inc}} \quad \text{on} \quad \Gamma_{\text{sph}}$$

where u_{inc} is the incident plane wave. The computational domain is truncated by a homogeneous impedance condition on the outer boundary of the cube Γ_{cube}.

$$p = u \cdot n \quad \text{on} \quad \Gamma_{\text{cube}}$$

[1]The author would like to thank Brendan Keith for all his help on constructing the geometry files for this simulation.

We solve the problem using our multigrid technology in the uniform refinement setting. The fine grid consists of 40960 hexahedra of quartic polynomial order. This results in a linear system of approximately 14 million degrees of freedom. The coarse grid is constructed by two h-coarsening steps of the fine grid. It consists of 640 quartic hexahedra and the linear system size is approximately 200 thousands degrees of freedom. The conjugate gradient algorithm starts with zero initial guess and terminates when the residual is less than 10^{-5}. We use a total of 10 smoothing iterations in each level and the smoother relaxation parameter is chosen to be $\theta = 0.2$. In this setting the preconditioned conjugate gradient algorithm converges in 12 iterations.

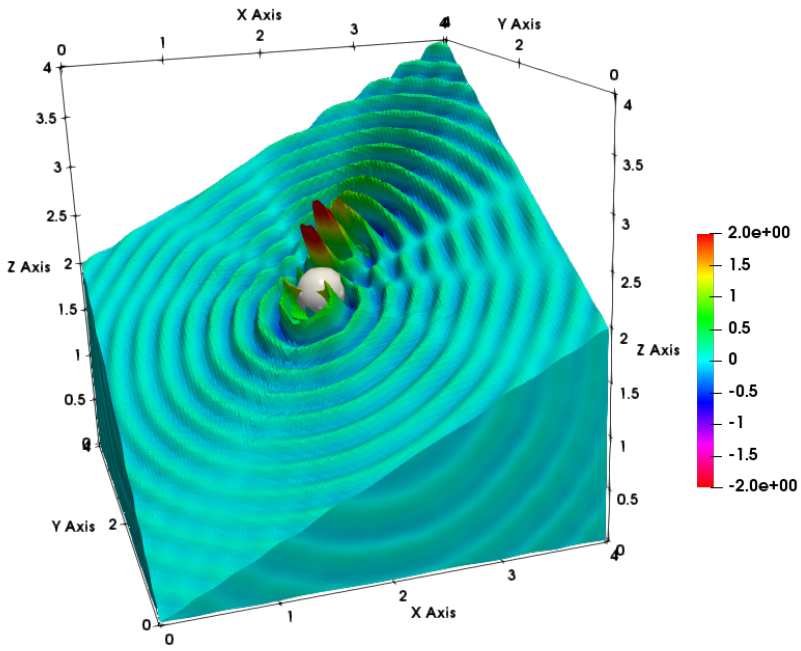


Figure 7.9: Scattered wave: real part of pressure

We emphasize that we attempted to solve the problem with a sparse multi-frontal solver but it was not possible because of high memory requirements ($\approx 350Gb$). The solution is shown in Figure 7.9 below. Note that we show the solution on the part of the domain below the plane defined by the point $(0.5, 0.5, 0.5)$ and the normal vector $(-1, -1, 2)$.

7.2.2 Plane wave scattering from a cube

Our second acoustics example involves scattering of a plane wave by a cube. The computational domain is $\Omega = (\frac{1}{7}, \frac{6}{7})^3 \backslash ((\frac{3}{7}, \frac{4}{7})^3$. The incident wave has low to medium frequency of $\omega = 16\pi$. We consider two cases for the direction of propagation, $(1, 0, 0)$ and $(1, 1, 0)$. In both cases the domain is truncated by a *Perfectly Matched Layer* (PML) region of length $L = \frac{1}{7}$ in each direction (see Figure 7.10). The construction of the PML is based on the work described in [115]. Additional details on how the DPG formulation is modified inside the PML region are given in Appendix D.

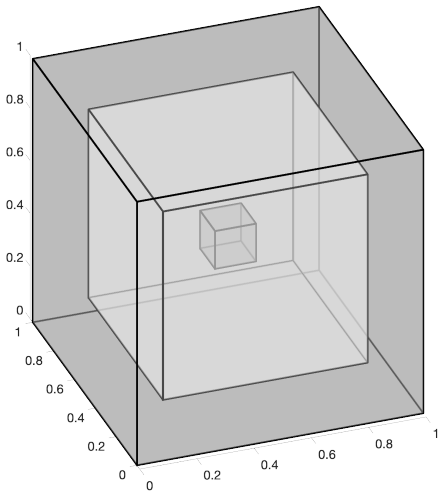


Figure 7.10: Domain including the PML region

We start the simulation with a mesh consisting of 342 cubes of size $h = \frac{1}{7}$ and polynomial order $p = 3$. Note that the initial mesh is fine enough to control the best approximation and the pollution effect. However, the singularities on the scatterer and the exponential decay of the wave in the PML region are not resolved and therefore the quality of the solution is not good. Starting automatic h-adaptivity, we anticipate that the DPG method will perform extensive refinements in order to resolve the singularities at the corners and edges of the cubic scatterer. Additionally, some refinements are expected to occur at the transition from the computational domain into the PML region.

The multigrid preconditioner setup is the same as in the previous examples. Here we use at most 6 multigrid levels and the stopping criterion of the CG solver is set to 10^{-5}. The sequence of meshes along with the solutions are shown in Figures 7.11 to 7.14. Notice that the meshes and the solutions are shown only in the part of the domain where $z \leq 0.5$. As expected, refinements occur close to the singularities and in the PML region. Observe how the solution changes as the singularities are resolved. Convergence results for the DPG residual and iteration counts for the preconditioned CG solver are given in Figures 7.15 and 7.16. As in the previous numerical examples the solver shows robust convergence throughout the adaptive refinement process.

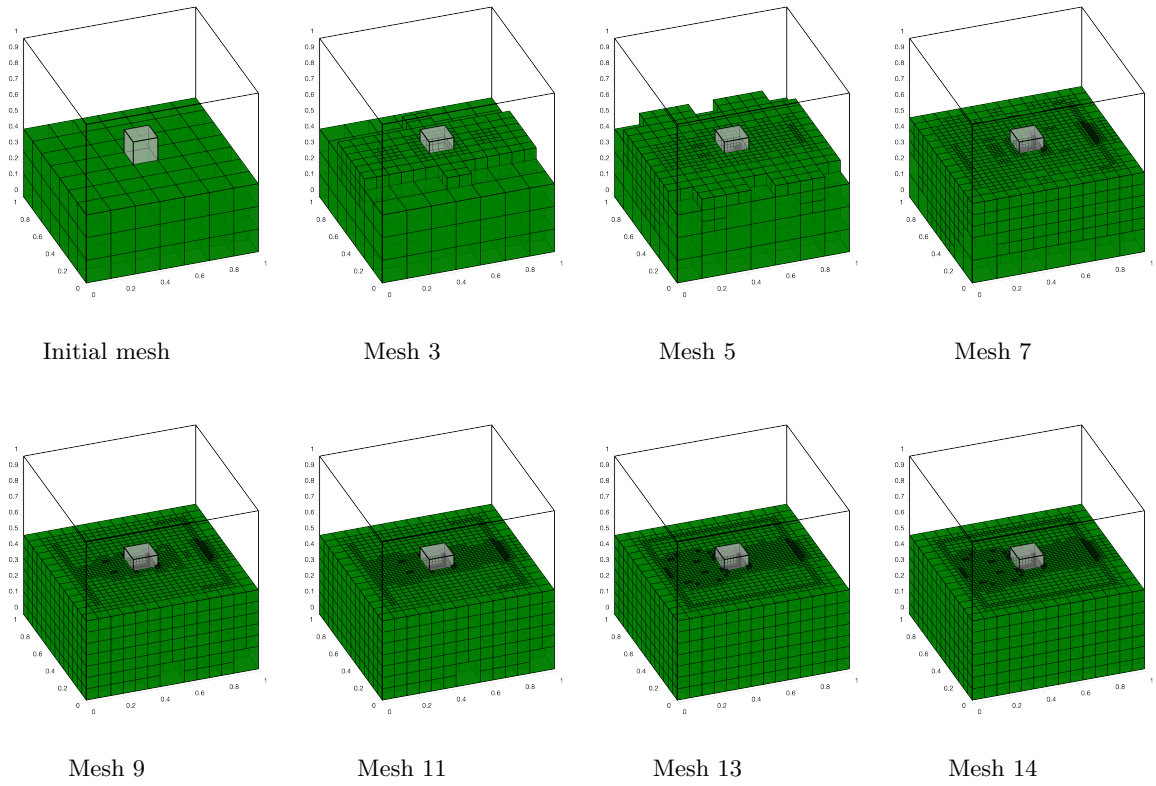


Figure 7.11: Wave propagating in the direction (1,0,0). Evolution of the h-adaptive mesh. As expected, a lot of refinements occur in the region close to the scatterer because the singularities have to be resolved. Additional refinements occur in the PML region

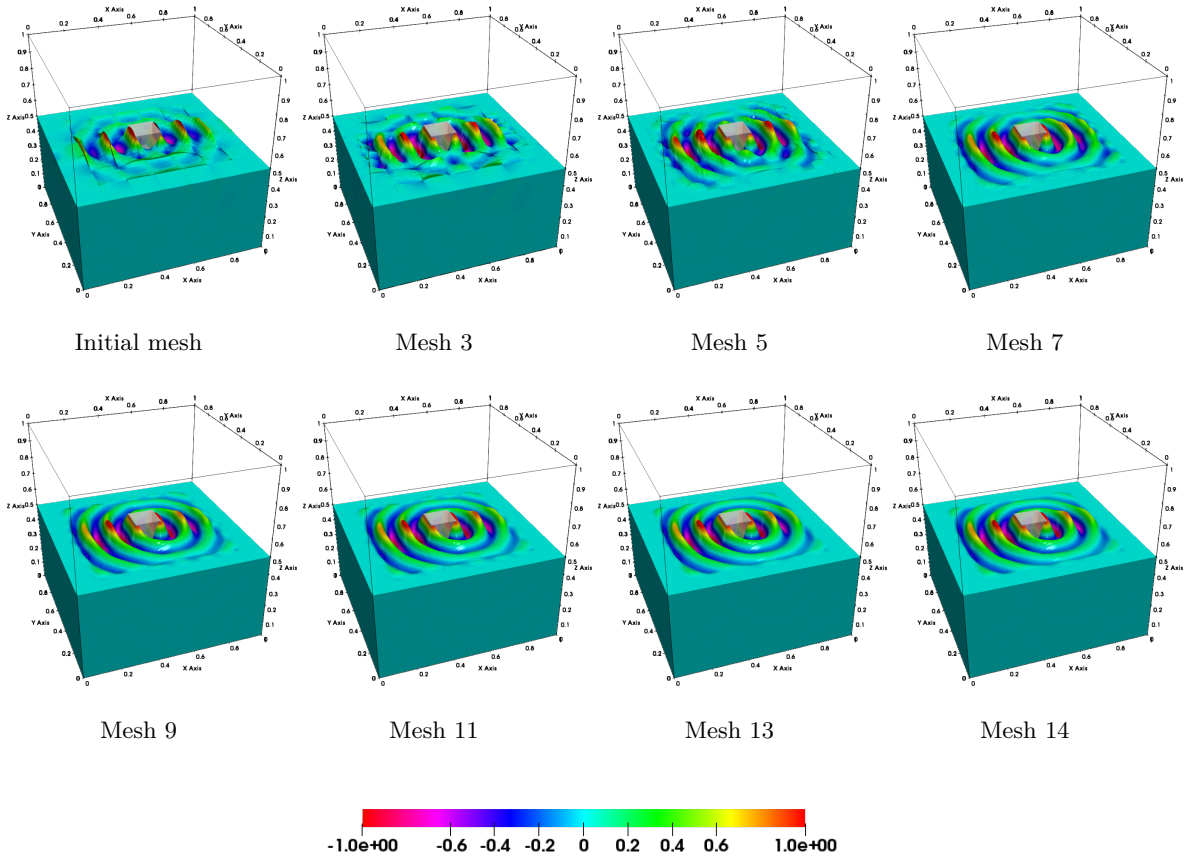
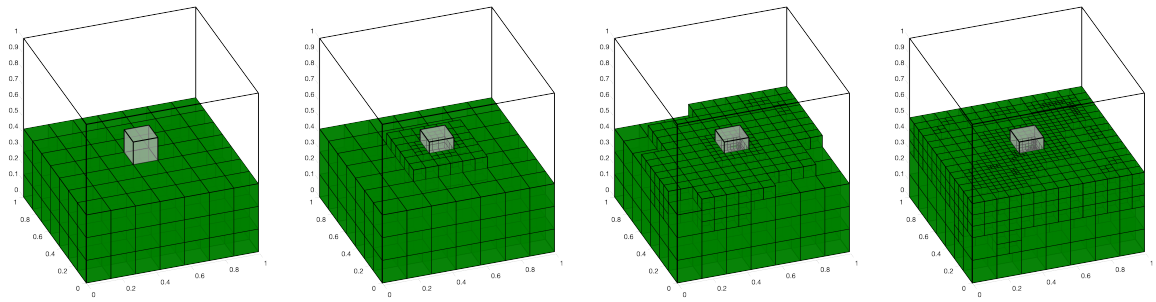
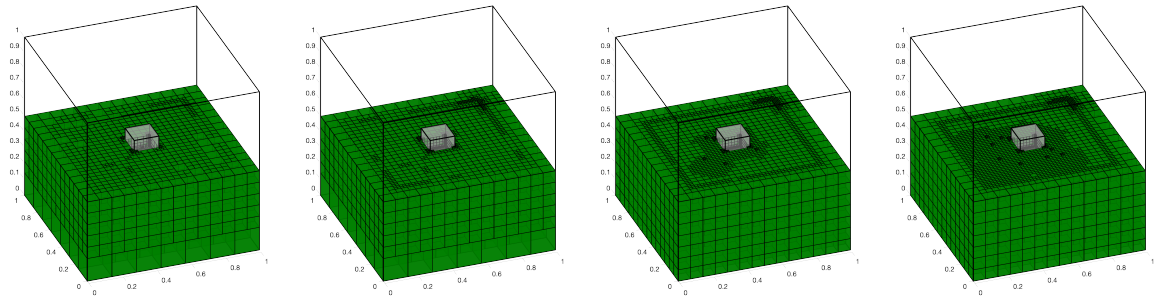


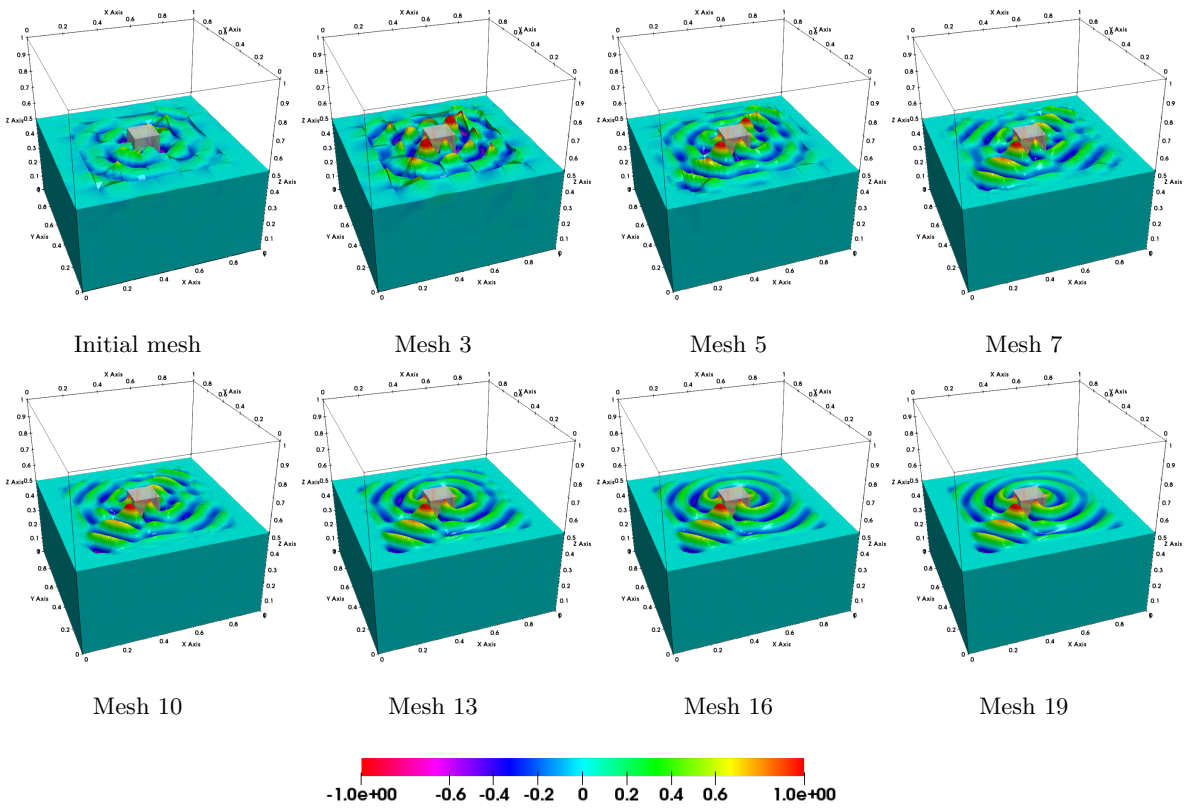
Figure 7.12: Wave propagating in the direction (1,0,0). Evolution of the solution. The solution rapidly decays in the PML region. Notice that the quality of the solution is affected by the resolution of the singularities





Mesh 10 Mesh 13 Mesh 16 Mesh 19

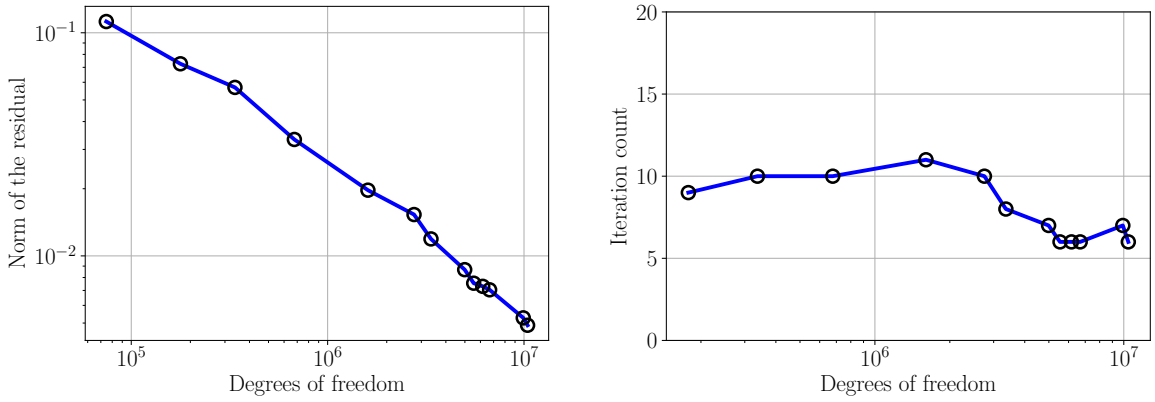
Figure 7.13: Wave propagating in the direction (1,1,0). Evolution of the h-adaptive mesh. As expected, a lot of refinements occur in the region close to the scatterer because the singularities have to be resolved. Additional refinements occur in the PML region



Initial mesh Mesh 3 Mesh 5 Mesh 7

Mesh 10 Mesh 13 Mesh 16 Mesh 19

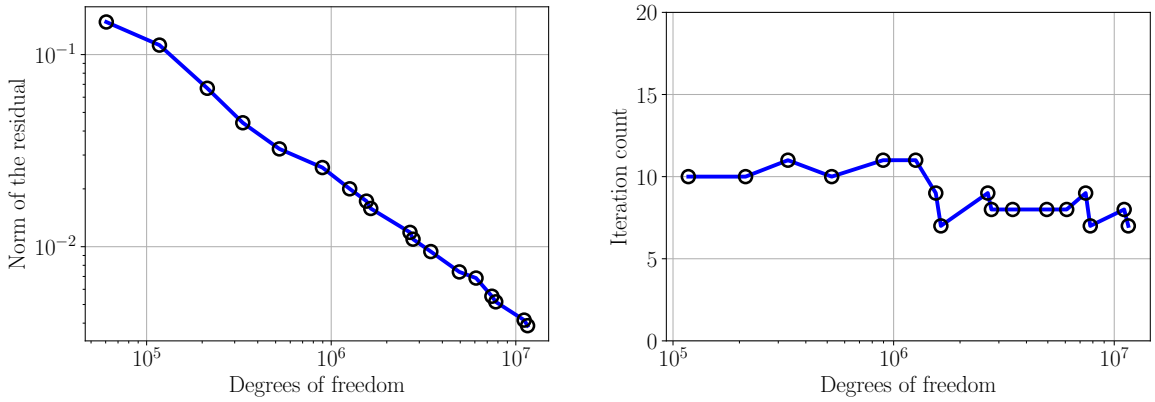
Figure 7.14: Wave propagating in the direction (1,1,0). Evolution of the solution. The solution rapidly decays in the PML region. Notice that the quality of the solution is affected by the resolution of the singularities



(a) Residual convergence

(b) CG iteration count

Figure 7.15: Residual and preconditioned CG convergence. Plane wave scattering from a cube - direction of propagation: (1,0,0).



(a) Residual convergence

(b) PCG iteration count

Figure 7.16: Residual and preconditioned CG convergence. Plane wave scattering from a cube - direction of propagation: (1,1,0).

7.2.3 Gaussian beam scattering from a cube

Our last experiment involves the simulation of a high frequency Gaussian beam scattering from a cube. The computational domain and the PML region are defined as in Section 7.2.2.

We run the simulation with angular frequency $\omega = 140\pi$. This corresponds to about 85 wavelengths inside the computational domain. We begin the simulation with a uniform mesh consisting of 342 cubes of polynomial order $p = 3$. We then perform adaptive hp-refinements with the following strategy. An element inside the computational domain is h-refined up to the point where its maximum side size becomes smaller than half a wavelength. An element with size smaller than that is p-refined. An exception is made for the elements adjacent to the corners and edges of the cubic scatterer and the elements inside the PML region, where only h-refinements are allowed. The simulation is terminated when the DPG residual reduces by an order of magnitude.

The iterative solver setup is as follows. The initial iterate consists of the degrees of freedom corresponding to the prolongation of the solution on the previous mesh to the current mesh. We perform 10 smoothing steps at each multigrid level and we choose the relaxation parameter to be $\theta = 0.2$. Additional computational time is saved by omitting to smooth in areas of the domain where there is no wave activity (the local patch residual is close to zero). Finally, in the simulation we use at most eight multigrid levels and the CG iterations are terminated when the l^2-norm of the residual becomes less than 10^{-3}.

In Figures 7.17 and 7.18 we show the sequence of meshes and the corresponding numerical solutions for the real part of the pressure respectively. For demonstration purposes a partial region of the mesh is shown, constructed by the union of the regions below three planes. These planes are defined by the point $0.5, 0.5, 0.5)$ and the normal vectors $(-0.5, -0.5, 1)$, $(0.5, -0.5, 1)$ and $(-0.5, 0.5, 1)$. Notice that the singularities at the corners and edges of the cubic scatterer have to be resolved before the wave can propagate. When the wave reaches the PML region, additional refinements occur in order to capture the steep decay. Convergence results are presented in Figure 7.19. Observe in Figure 7.19a that the norm of DPG residual, which drives the adaptive refinements, tends to zero only after the decay in the PML region is resolved. Lastly, iteration counts for the CG solver preconditioned with our multigrid technology are given in Figure 7.19b. Notice that the number of iterations remains under control throughout the adaptive process.

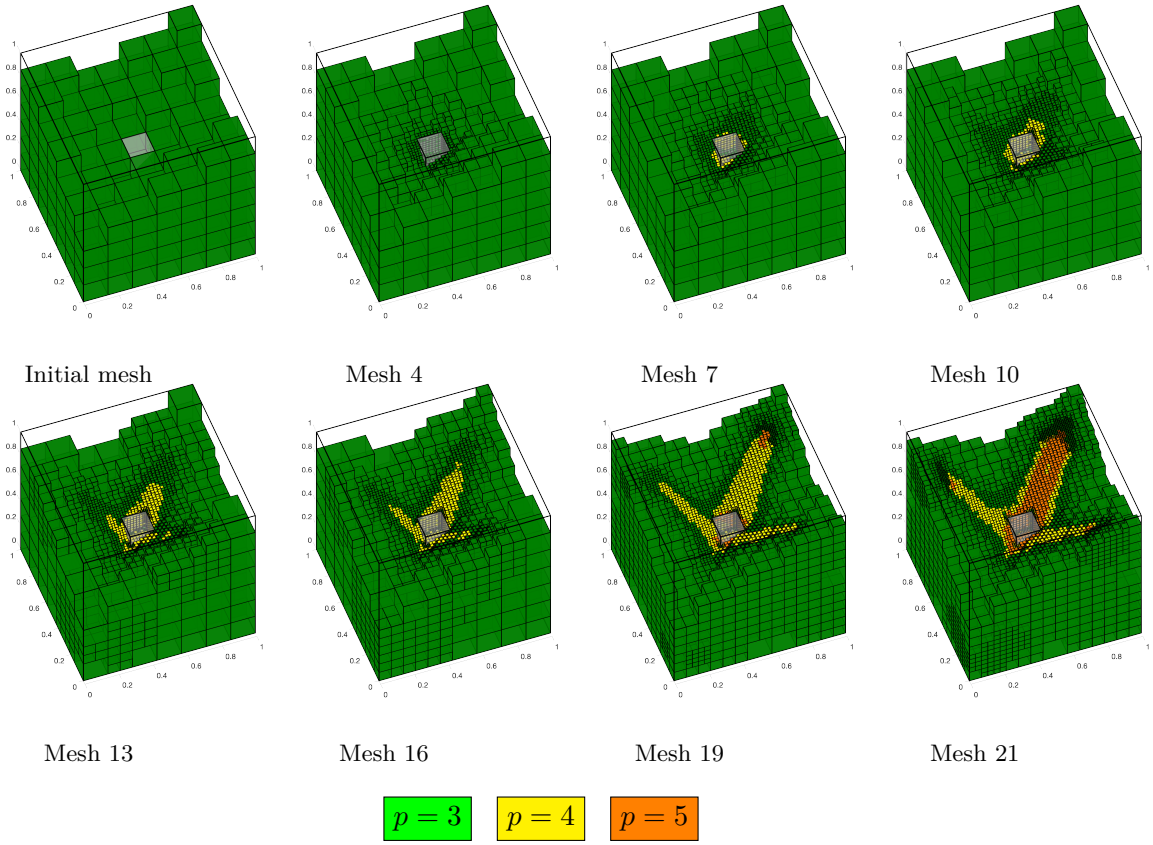
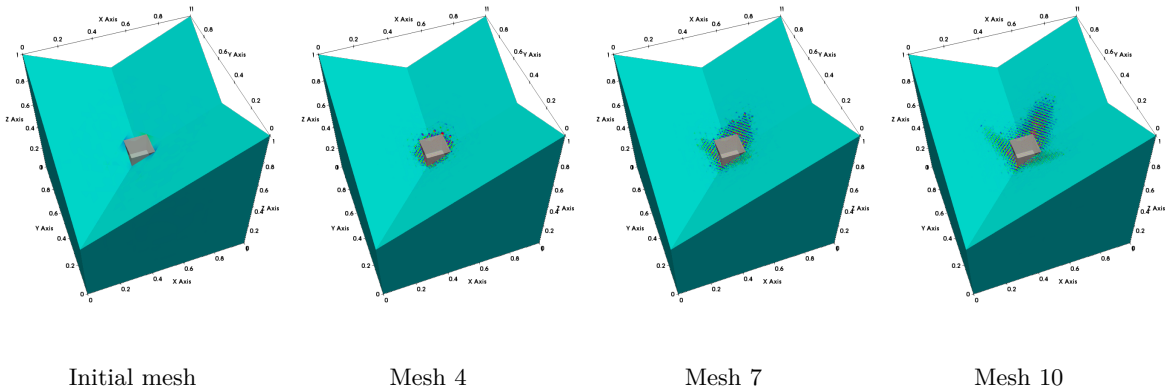


Figure 7.17: Evolution of the hp-adaptive mesh. Notice that the singularities at the scatterer have to be resolved before the wave can propagate.



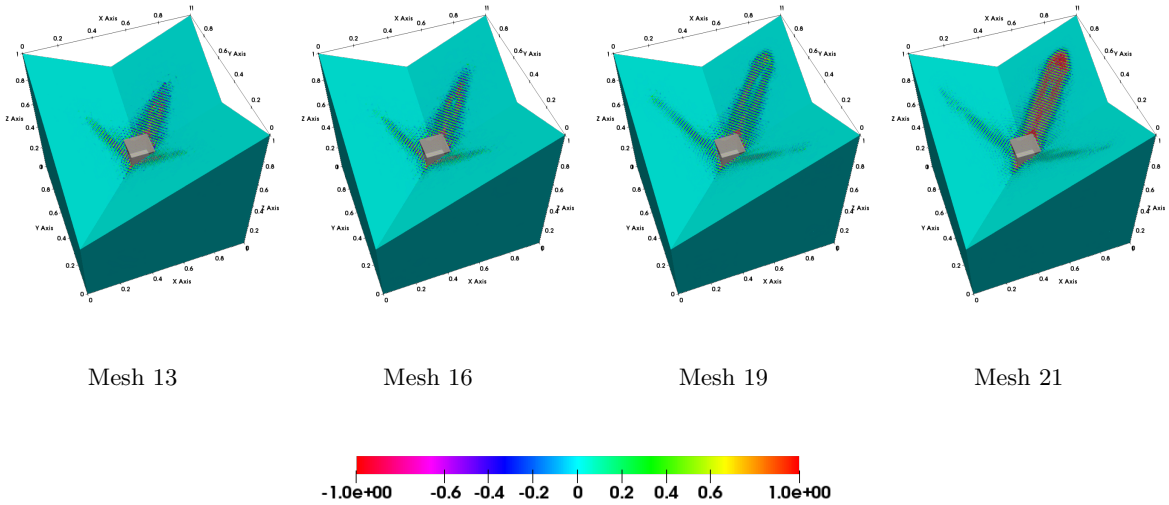
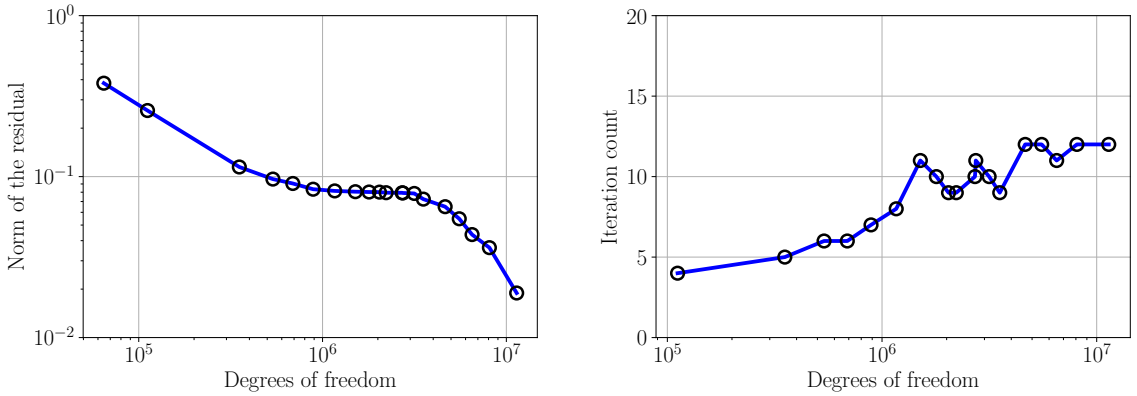


Figure 7.18: Evolution of the solution. Here the real part of the acoustic pressure is displayed.



(a) Residual convergence

(b) Iteration count

Figure 7.19: Convergence of the DPG residual and the preconditioned CG solver. Note that the number of iterations of the iterative solver is controlled throughout the adaptive process.

Chapter 8

Conclusion

8.1 Work summary

The main accomplishment of this work is the development of a general iterative solution scheme for DPG linear systems and its application to the solution of challenging problems in the area of wave propagation. The construction is heavily based on the well established multigrid theory but also the numerous attractive features of the DPG method.

Since the DPG method always delivers symmetric (Hermitian) positive definite linear systems, the Conjugate Gradient solver can be employed for their solution. Consequently, construction and analysis of preconditioners are based on the subspace correction theory of Xu [119, 117, 118]. This provides additional comfort and trust on the iterative solver, since convergence to the true solution is always guaranteed. Theoretical results on our construction were developed for the one-level setting and presented in Chapter 4.

The unconditional pre-asymptotic discrete stability provided by the DPG method, combined with the DPG built-in error indicator, allow for automatic adaptive mesh refinements starting from very coarse meshes. For problems with localized solutions, this turned out to be very crucial, since the computational resources can be correctly assigned only in areas of the domain that there is wave activity. The key point, making this work successful, was the integration of the iterative solver within the adaptive refinement process. We demonstrated that adaptive mesh refinements can be driven by partially converged solutions, produced by the preconditioned CG solver. The iterative solver was accelerated by the multigrid preconditioner for adaptive *hp*-meshes using the history of refinements. We emphasize that the construction was not standard, i.e., the underlying linear system involved only interface unknowns, and therefore construction of the inter-grid transfer operators included additional operators based

on Schur complements.

While this technology can be used in a wide range of applications, emphasis was given on wave propagation problems in the high frequency regime. Several numerical experiments in both two and three space dimensions were presented for acoustics and Maxwell equations. With our multigrid technology we were able to efficiently solve "large" problems, which otherwise would be computationally intractable. Even though our theoretical work was limited to the one-level setting, numerical results on the multigrid preconditioner displayed in Chapter 7, suggest that under certain circumstances, the solver converges uniformly with respect to the mesh size, the polynomial order and the frequency.

8.2 Future directions

Driven by several engineering problems within our research group, a possible future direction involves an implementation of the presented multigrid technology for distributed memory computer architectures using MPI. In particular, an undergoing project includes the simulation of optical laser amplifiers [97], a problem which requires sufficient resolution of waves of hundreds of thousands of wavelengths. An MPI implementation of our hp3D code is currently under development. The multigrid technology will be used along with automatic adaptivity in order to simulate the so called Transverse Mode Instability (TMI), a phenomenon characterized by sudden degradation of the laser beam quality [97]. Since, direct solvers, fail to scale efficiently on modern many-core architectures, an efficient iterative solver seems to be the only alternative choice.

Finally, our theoretical analysis on the one-level additive preconditioner for the acoustics problem can serve as a stepping stone on a further study in the multilevel setting. While several other theoretical works exist on preconditioning DPG systems, analytical results on preconditioning wave operators are yet to be explored.

Appendices

Appendix A

Construction of DPG Fortin operators

In this chapter we outline, without going into much detail our work on the construction of Fortin operators for the H^1 and $H(\mathrm{div})$ case. For further reading we refer the reader to our paper [98]. The purpose of this construction is to quantify the loss of stability when moving from the ideal to the practical DPG method.

A.1 Outline of the construction

Recall the Petrov–Galerkin scheme for the ideal DPG case given by (2.7):

$$\begin{cases} u_h \in U_h \subset U \\ b(u_h, v_h) = l(v_h), \quad v_h \in V_h^{\mathrm{opt}} := \mathrm{T}(U_h) \end{cases}$$

where the ideal trial-to-test operator $\mathrm{T} : U_h \to V$ is defined by

$$(\mathrm{T}\delta u_h, \delta v)_V = b(\delta u_h, \delta v), \quad \delta u_h \in U_h,\, \delta v \in V.$$

Here, U and V are Hilbert spaces, $U_h \subset U$ is a finite dimensional subspace of U, $b(\cdot,\cdot)$ is a continuous bilinear form defined on the product $U \times V$ and $l(\cdot)$ is a continuous linear functional defined on V. As demonstrated in Section 2.2, computing V_h^{opt} involves the solution of an infinite dimensional problem. Hence, practically a truncated finite dimensional subspace $V^r \subset V$ is considered and the ideal trial-to-test operator is approximated by the operator T^r defined by

$$(\mathrm{T}^r u_h, v)_V = b(u_h, v), \quad v \in V^r.$$

This leads to the practical DPG problem:

$$\begin{cases} u_h \in U_h \subset U \\ b(u_h, v_h) = l(v_h), \quad v_h \in V_r^{\mathrm{opt}} = \mathrm{T}^r(U_h). \end{cases}$$

A stability estimate for the practical DPG method can be derived under the assumption of the existence of a Fortin operator. Let $\Pi : V \to V^r$ be a linear map satisfying

$$\|\Pi v\| \leq \|\Pi\|\|v\|, \quad v \in V$$
$$b(w_h, v - \Pi v) = 0, \qquad w_h \in U_h,\, v \in V.$$

Then the following stability estimate for the DPG method holds

$$\|u - u_h\|_U \leq \frac{M}{\gamma}\|\Pi\| \inf_{w_h \in U_h} \|u - w_h\|_U$$

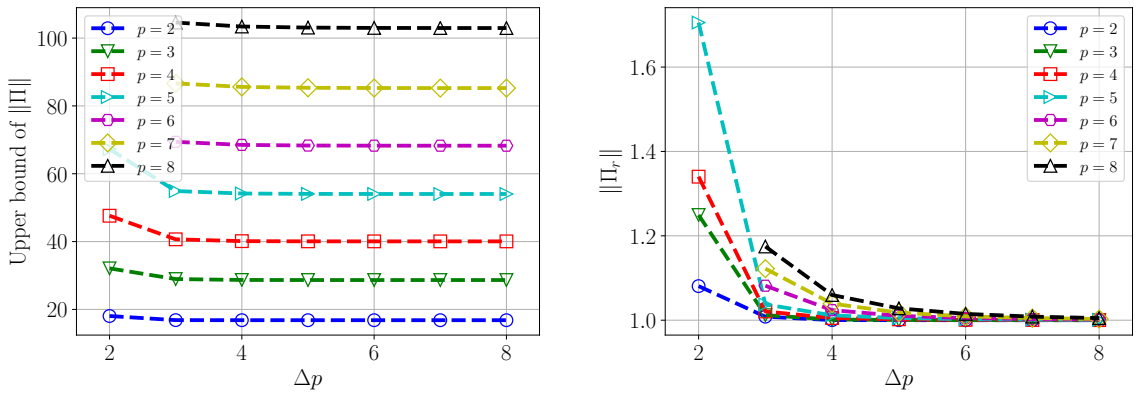
where $\|\Pi\|$ is the operator norm of Π. A rigorous theoretical analysis of the practical DPG method is demonstrated in [65]. The work in [98] investigates how stability is altered by the norm $\|\Pi\|$, i,e., what is the dependence of the $\|\Pi\|$ on the polynomial order p and the enriched test space order Δp. The construction is done on a two dimensional triangular mesh. The key point is that the use of broken test space and some scaling arguments allow for the construction to be reduced on a single master element.

Two approaches are followed. This first one entails an auxiliary problem resulting from a set of sufficient conditions. An upper bound of the continuity constant of $\Pi : V \to V^r$ is then estimated in terms of the inf-sup condition of the auxiliary problem. This leads to a rather pessimistic upper bound of $\mathcal{O}(10) - \mathcal{O}(10^2)$. The second approach, involves the construction of a sequence of approximate Fortin operators $\Pi_{\Delta r} : V^{r+\Delta r} \to V^r$ and exact computation of their continuity constants. This gives an optimistic lower bound of $\mathcal{O}(1)$.

A.2 Numerical results

Numerical results are displayed in Figures A.1 and A.2. In both H^1 and $H(\text{div})$ construction, an increase of the upper bound can be observed when increasing p. This is expected since for higher p, resolution of the optimal test functions becomes more difficult. Increasing Δp increases stability only by a marginal factor. We emphasize however that the construction is based on sufficient but not necessary conditions, and therefore the obtained upper bound might be very loose. On the other hand, rapid convergence is observed when increasing Δp for

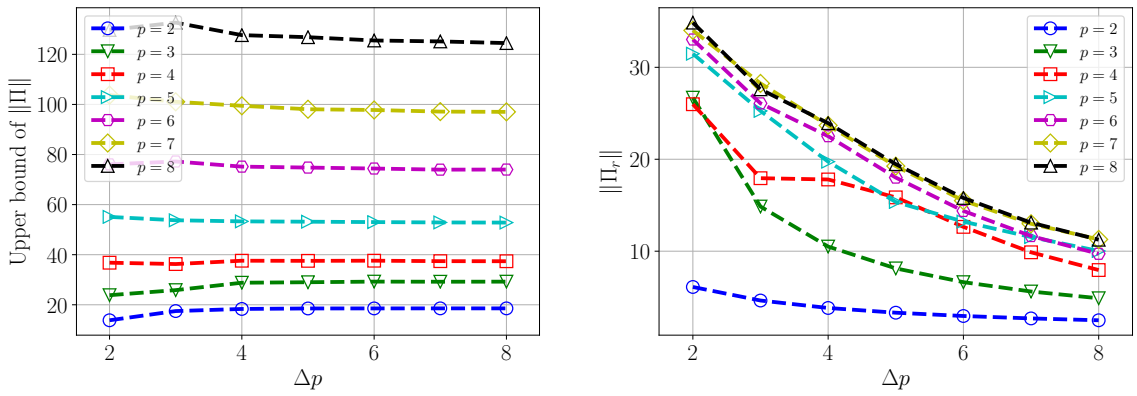
the exact computation of the norm of the approximate Fortin operators, indicating minimal loss of stability.



(a) Upper bound of $\|\Pi\|$

(b) Exact norm of $\Pi_{\Delta r}$

Figure A.1: H^1 Fortin operator construction



(a) Upper bound of $\|\Pi\|$

(b) Exact norm of $\Pi_{\Delta r}$

Figure A.2: $H(\mathrm{div})$ Fortin operator construction

Appendix B

A new discrete least squares (DLS) approach for DPG systems

In this chapter, we discuss a new methodology for solving the DPG system. This new procedure involves an alternate assembly algorithm, where the global stiffness matrix is rectangular and the linear system is overdetermined. The solution of the system, is then obtained using orthogonalization algorithms (e.g. QR factorization). Although, this approach often leads to slower solution times and increased memory requirements, it is suitable for ill-conditioned systems, since now the condition number of the matrix grows with $\mathcal{O}(h^{-1})$. We describe the main ideas behind this methodology and present selective numerical results that support our claim. We refer the reader to [83] for further discussion on the subject and more examples.

B.1 Description of the method

Recall the minimum residual formulation (2.10) for the practical DPG method:

$$u_h = \arg \min_{w_h \in U_h} \|\mathrm{R}_{V^r}^{-1}(l - \mathrm{B}w_h)\|_V \tag{B.1}$$

and let $\mathfrak{U}_\mathfrak{h} = \{\mathfrak{u}_\mathfrak{i}\}_{i=1}^N$ and $\mathfrak{V}_\mathfrak{r} = \{\mathfrak{v}_\mathfrak{i}\}_{i=1}^M$ (where $M > N$) denote bases for the discrete trial space U_h and test space V_r respectively. We can now discretize all the finite dimensional operators in (B.1), and arrive at a discrete least square problem.

Indeed, let $\mathsf{B}_{ij} = b(\mathfrak{u}_j, \mathfrak{v}_i)$, $\mathsf{G}_{ij} = (\mathfrak{v}_i, \mathfrak{v}_j)_V$, $\mathsf{l} = l(\mathsf{v}_i)$ and denote by $\mathbb{F}$ to be the field ($\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$). Then, we want to find the set of coefficients $\mathsf{w} = [\mathsf{w}_i]_{i=1}^N \in \mathbb{F}^N$ such that the solution

$$\mathfrak{u}_h = \sum_{i=1}^N \mathsf{w}_i \mathfrak{u}_i$$

satisfies the discrete minimization problem:

$$\mathsf{u} = \arg\min_{\mathsf{w}\in\mathbb{F}^N} (\mathsf{l} - \mathsf{B}\mathsf{w})^* \mathsf{G}^{-1} (\mathsf{l} - \mathsf{B}\mathsf{w}) \tag{B.2}$$

where $\mathsf{u} = [\mathsf{u}_i]_{i=1}^N \in \mathbb{F}^N$.

The matrix G is Hermitian (symmetric) positive definite and block diagonal. Therefore, computing its Cholesky factorization is computationally negligible. If we write $\mathsf{G} = \mathsf{L}\mathsf{L}^*$ then from (B.2) we have:

$$\begin{aligned} \mathsf{u} &= \arg\min_{\mathsf{w}\in\mathbb{F}^N} (\mathsf{l} - \mathsf{B}\mathsf{w})^* (\mathsf{L}\mathsf{L}^*)^{-1} (\mathsf{l} - \mathsf{B}\mathsf{w}) \\ &= \arg\min_{\mathsf{w}\in\mathbb{F}^N} \left(\mathsf{L}^{-1}(\mathsf{l} - \mathsf{B}\mathsf{w})\right)^* \left(\mathsf{L}^{-1}(\mathsf{l} - \mathsf{B}\mathsf{w})\right) \\ &= \arg\min_{\mathsf{w}\in\mathbb{F}^N} \|\mathsf{L}^{-1}(\mathsf{l} - \mathsf{B}\mathsf{w})\|_2^2 \end{aligned} \tag{B.3}$$

Define $\widetilde{\mathsf{B}} = \mathsf{L}^{-1}\mathsf{B}$ and $\widetilde{\mathsf{l}} = \mathsf{L}^{-1}\mathsf{l}$. Then the solution of (B.3) is given by the overdetermined system of equations

$$\widetilde{\mathsf{B}}\mathsf{u} \overset{LS}{=} \widetilde{\mathsf{l}} \tag{B.4}$$

where $\overset{LS}{=}$ is understood in the discrete least squares sense.

We can now solve (B.4) directly, by employing orthogonalization algorithms, such as the QR-factorization, or we can form the normal equations instead (see (2.14)) and solve with Cholesky decomposition algorithms. To date, the latter approach has been the most common procedure for solving the DPG linear system. For well-conditioned problems, the normal equations approach seems to be the ideal choice, since there exist time- and memory-efficient sparse linear solvers for the solution (e.g. multi-frontal solvers). However, the condition number of the matrix is practically squared compared to the condition number of the original rectangular stiffness matrix. When it comes to the use of iterative solvers, their convergence is influenced by the condition number. Additionally, for nearly ill-condition problems, even the direct solvers might fail, where in such a case considering the overdetermined system is the only alternative [102].

B.2 Static condensation for the overdetermined system

Static condensation is standard practice for square systems. The degrees of freedom, associated with the interior nodes of an element are eliminated out of the system. In practice, independent blocks within the global matrix are inverted and removed using a Schur complement technique. Then, one solves a modified but significantly smaller system that involves only the interface unknowns (i.e, the coefficients of the shape functions (degrees of freedom) that their support extends to more that one element). In this section, we present an analogous procedure for the overdetermined system (B.4).

In the minimization problem (B.3) we consider a splitting of the coefficients w into interface and bubble components, i.e, $\mathsf{w} = [\mathsf{w}_{\text{interf.}}|\mathsf{w}_{\text{bubb.}}]^T$. Similarly $\mathsf{u} = [\mathsf{u}_{\text{interf.}}|\mathsf{u}_{\text{bubb.}}]^T$ and $\widetilde{\mathsf{B}} = [\widetilde{\mathsf{B}}_{\text{interf.}}|\widetilde{\mathsf{B}}_{\text{bubb.}}]^T$. Then, the minimizer of (B.4) can be found by considering two separate minimization problems. That is

$$\begin{aligned}\min_{\mathsf{w}\in\mathbb{F}^N} \|\widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}\mathsf{w}\|_2^2 &= \min_{\mathsf{w}_{\text{interf.}}\in\mathbb{F}^{N_{\text{interf.}}}} \min_{\mathsf{w}_{\text{bubb.}}\in\mathbb{F}^{N_{\text{bubb.}}}} \|\widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}[\mathsf{w}_{\text{interf.}}|\mathsf{w}_{\text{bubb.}}]^T\|_2^2 \\ &= \min_{\mathsf{w}_{\text{bubb.}}\in\mathbb{F}^{N_{\text{bubb.}}}} \min_{\mathsf{w}_{\text{interf.}}\in\mathbb{F}^{N_{\text{interf.}}}} \|\widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}[\mathsf{w}_{\text{interf.}}|\mathsf{w}_{\text{bubb.}}]^T\|_2^2\end{aligned}$$

Assuming now that $\mathsf{w}_{\text{interf.}}$ is fixed, we can minimize for the bubble coefficients, i.e,

$$\mathsf{u}_{\text{bubb.}} = \underset{\mathsf{w}_{\text{bubb.}}\in\mathbb{F}^{N_{\text{bubb.}}}}{\arg\min} \|(\widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}_{\text{interf.}}\mathsf{w}_{\text{interf.}}) - \widetilde{\mathsf{B}}_{\text{bubb.}}\mathsf{w}_{\text{bubb.}}\|_2^2$$

or

$$\mathsf{u}_{\text{bubb.}} = \widetilde{\mathsf{B}}^{\dagger}_{\text{bubb.}}(\widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}_{\text{interf.}}\mathsf{w}_{\text{interf.}}) \tag{B.5}$$

where $\widetilde{\mathsf{B}}^{\dagger}_{\text{bubb.}} = (\widetilde{\mathsf{B}}^{*}_{\text{bubb.}}\widetilde{\mathsf{B}}_{\text{bubb.}})^{-1}\widetilde{\mathsf{B}}^{*}_{\text{bubb.}}$ denotes the pseudo-inverse of $\widetilde{\mathsf{B}}_{\text{bubb.}}$. We can now derive a minimization only for the interface coefficients by substituting (B.5) into (B.4). Indeed, (B.4) can be written as

$$\widetilde{\mathsf{B}}_{\text{interf.}}\mathsf{w}_{\text{interf.}} + \widetilde{\mathsf{B}}_{\text{bubb.}}\mathsf{w}_{\text{bubb.}} \overset{LS}{=} \widetilde{\mathsf{l}}$$

and therefore for fixed $\mathsf{w}_{\text{bubb.}} = \mathsf{u}_{\text{bubb.}}$ we have

$$\widetilde{\mathsf{B}}_{\text{interf.}}\mathsf{w}_{\text{interf.}} + \widetilde{\mathsf{B}}_{\text{bubb.}}\widetilde{\mathsf{B}}^{\dagger}_{\text{bubb.}}(\widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}_{\text{interf.}}\mathsf{w}_{\text{interf.}}) \overset{LS}{=} \widetilde{\mathsf{l}}.$$

Finally, $\mathsf{u}_{\text{interf.}}$ is obtained by solving the least squares problem

$$(\mathrm{I} - \widetilde{\mathsf{B}}_{\text{bubb.}}\widetilde{\mathsf{B}}^{\dagger}_{\text{bubb.}})\widetilde{\mathsf{B}}_{\text{interf.}}\mathsf{w}_{\text{interf.}} \overset{LS}{=} \widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}_{\text{bubb.}}\widetilde{\mathsf{B}}^{\dagger}_{\text{bubb.}}\widetilde{\mathsf{l}} \tag{B.6}$$

Remark B.1. In practice the matrix $\widetilde{\mathsf{B}}_{\text{bubb.}}$ consists of independent rectangular blocks. We can therefore compute its QR-factorization very efficiently by considering one element at a time. By using the QR-factorization, we can avoid forming the pseudo-inverse $\widetilde{\mathsf{B}}^{\dagger}_{\text{bubb.}}$, which potentially can cause conditioning issues. Indeed, let $\mathsf{Q}_{\text{bubb.}}$ and $\mathsf{R}_{\text{bubb.}}$ be the QR factors of $\widetilde{\mathsf{B}}_{\text{bubb.}}$, i.e, $\widetilde{\mathsf{B}}_{\text{bubb.}} = \mathsf{Q}_{\text{bubb.}}\mathsf{R}_{\text{bubb.}}$, where $\mathsf{Q}_{\text{bubb.}}$ is unitary and $\mathsf{R}_{\text{bubb.}}$ is upper-triangular. Then (B.5) and (B.6) reduce to

$$\mathsf{u}_{\text{bubb.}} = \mathsf{R}^{-1}_{\text{bubb.}}\mathsf{Q}_{\text{bubb.}}(\widetilde{\mathsf{l}} - \widetilde{\mathsf{B}}_{\text{interf.}}\mathsf{w}_{\text{interf.}}) \tag{B.7}$$

and

$$\underbrace{(\mathrm{I} - \mathsf{Q}_{\text{bubb.}}\mathsf{Q}^{*}_{\text{bubb.}})\widetilde{\mathsf{B}}_{\text{interf.}}}_{\mathsf{B}_{\text{con}}}\mathsf{w}_{\text{interf.}} \overset{LS}{=} (\mathrm{I} - \mathsf{Q}_{\text{bubb.}}\mathsf{Q}^{*}_{\text{bubb.}})\widetilde{\mathsf{l}} \tag{B.8}$$

respectively. The solution coefficients for the bubble basis functions can be retrieved by (B.7) after solving the global problem (B.8). Note that this operation involves only local computations and can be implemented in parallel.

Remark B.2. It can be easily verified that (B.8) is equivalent to the condensed system for the normal equations. For a detailed proof we refer the reader to [83]

Remark B.3. Recall that in the case of DPG the test space is broken (discontinuous). Therefore, the assembly of the global overdetermined system involves no accumulation, since there is no overlap between rows (see Figures B.1 and B.2). The assembly algorithms are described in detail in [83].

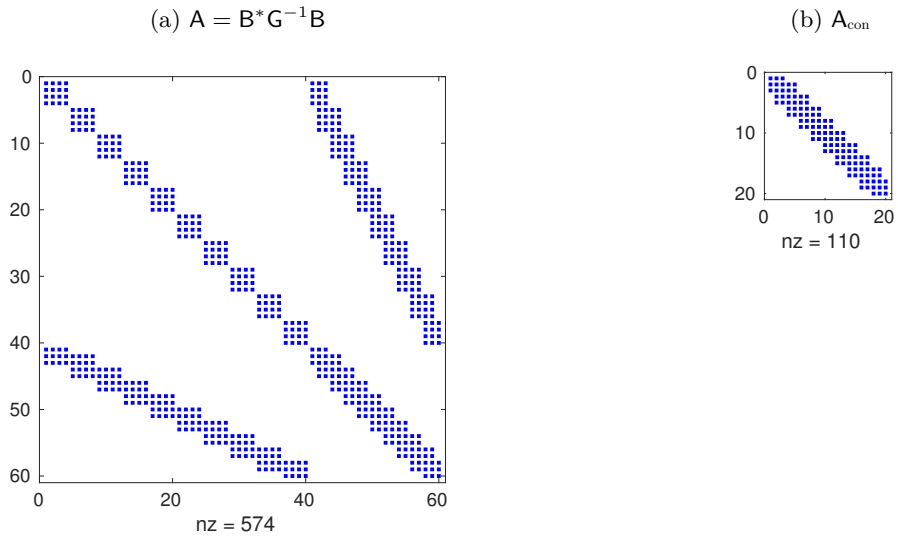


Figure B.1: DPG stiffness matrix for the normal equations approach. Here, A denotes the matrix of the total system and A_{con} the matrix for the condensed system.

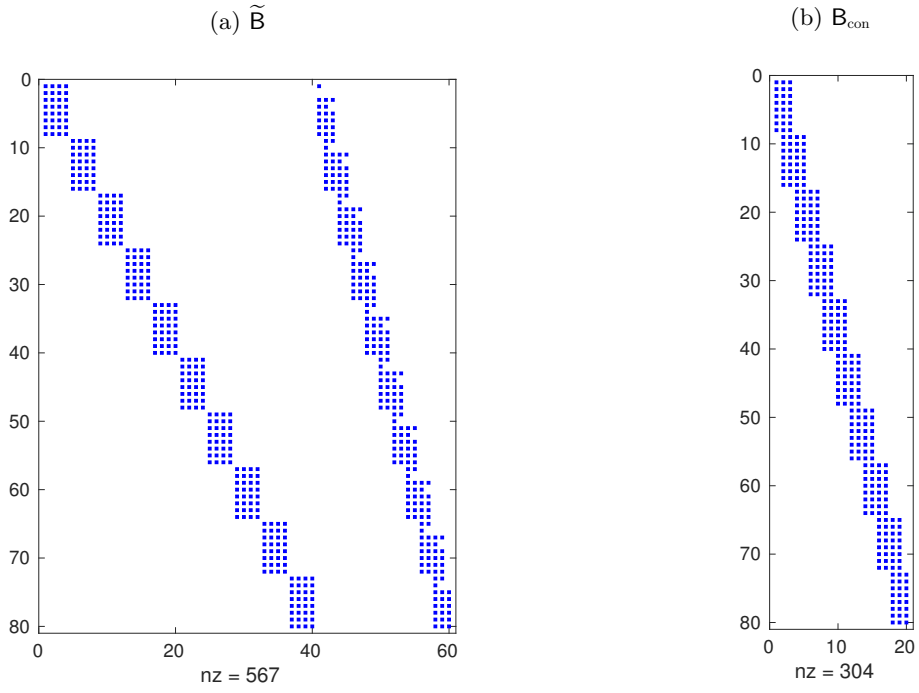


Figure B.2: DPG stiffness matrix for the overdetermined system. Here, $\widetilde{\mathsf{B}}$ denotes the rectangular matrix for the total system and B_{con} the matrix after static condensation.

B.3 A failure study

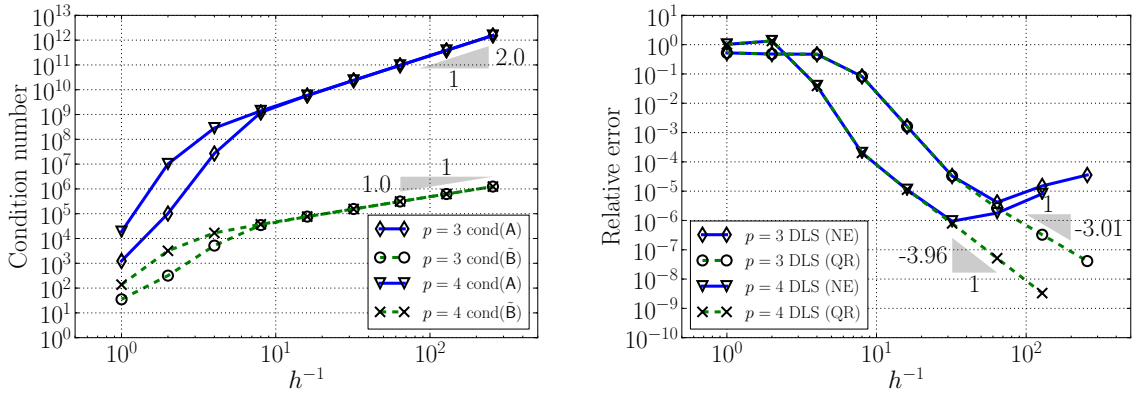
In order to demonstrate the effect of the roundoff error on the quality of the solution, and how it is related with the condition number, we consider the example of linear acoustics, similar to the one in Chapter 3. This time, we impose hard boundary conditions on the entire boundary, i.e, $u|_{\partial\Omega}\cdot n = g$. We choose to solve the problem at a near resonance frequency using the ultraweak formulation. Here, the global stiffness matrix would become ill-conditioned. In this case we seek a solution $(p,u,\hat{p},\hat{u}_n) \in L^2(\Omega)\times(L^2(\Omega))^d \times H^{1/2}(\partial\Omega_h)\times H_0^{-1/2}(\partial\Omega_h)$ such that

$$b\big((p,u,\hat{p},\hat{u}_n),(q,v)\big) = \ell\big((q,v)\big)\,, \quad (q,v)\in H^1(\Omega_h)\times H(\mathrm{div},\Omega_h)\,,$$

where

$$\begin{aligned} b\big((p,u,\hat{p},\hat{u}_n),(q,v)\big) =& -(p, i\omega q + \mathrm{div} v) - (u, i\omega v + \mathrm{grad}\, q) \\ &+ \langle \hat{u}_n, q\rangle_{\partial\Omega_h} + \langle \hat{p}, v\cdot n\rangle_{\partial\Omega_h} \\ \ell\big((q,v)\big) =& \ (f,q)_{\Omega_h} - \langle \widetilde{g}, q\rangle_{\partial\Omega_h}\,. \end{aligned}$$

and $\widetilde{g}\in H^{-1/2}(\partial\Omega_h)$ is an extension of $\hat{g}\in H^{-1/2}(\partial\Omega)$ to $\partial\Omega_h$.



(a) Condition number

(b) Relative error

Figure B.3: Linear acoustics near resonance (frequency $\omega = 0.5001\cdot\pi$). Here, A denotes the global stiffness matrix of the normal equations approach, i.e., $\mathsf{A} = \mathsf{B}^*\mathsf{G}^{-1}\mathsf{B}$.

We solve the problem for $\omega = 0.5001\cdot 2\pi$. Note that for $\omega = \pi$ the problem is ill-posed, since it is the first eigenvalue of the Laplace operator. We use the Gaussian beam as a

manufactured solution (see [105]), starting from a single element and performing eight uniform h-refinements. We report our results for order of approximation $p = 3$ and 4, in Figure B.3. We mention here, that for the solution of the normal equations and the overdetermined system, we used the packages MUMPS [88, 1] and QR_MUMPS [15] respectively. First, from Figure B.3a, we verify that the condition number for the normal equations is $\mathcal{O}(h^{-2})$. On the other hand, as expected, the condition number of the overdetermined system grows only linearly with h^{-1}. An interesting observation is that, in the asymptotic regime there is no significant difference in the condition numbers for the two different orders of approximation.

In Figure B.3b we plot the relative L^2 error for each refinement. Both approaches recover the theoretical rate of convergence. However, solving the normal equations leads to inaccurate solutions and loss of the convergence rates after the sixth mesh. Here, the effect of the roundoff error is very obvious. A careful look on the condition number plot, reveals this fact. For example, for $p = 4$ on the sixth mesh, the condition number of the normal equations is $\mathcal{O}(10^{10})$, and so we expect to lose about ten significant digits of accuracy. The corresponding error on this mesh is of the order of 10^{-6}. Therefore, this is the last reliable solution if double precision arithmetic is used (machine epsilon $\epsilon \approx 10^{-16}$). On the seventh mesh the condition number is $\mathcal{O}(10^{11})$ and the expected error is $\mathcal{O}(10^{-7})$. In this case too many significant digits are lost and the normal equations approach diverges. On the contrary the discrete least squares approach is able to maintain the optimal rates of convergence for two additional h-refinements.

Appendix C

Iterative Solvers

C.1 Iterator as a preconditioner

Theorem C.1. *Consider the linear iteration* (4.2) *and suppose* A *and* B *are self-adjoint with respect to* $(\cdot,\cdot)$*. If* B *is bounded, bijective, positive definite and*

$$\eta = \|\mathrm{I} - \mathsf{AB}\|_{\mathsf{B}} < 1 \tag{C.1}$$

then

1. A *is positive definite.*
2. *Iteration* (4.2) *is convergent.*
3. *The condition number* $\kappa(\mathsf{AB}) = \dfrac{\lambda_{\max}(\mathsf{AB})}{\lambda_{\min}(\mathsf{AB})}$ *satisfies* $\kappa(\mathsf{AB}) \leq \dfrac{1+\eta}{1-\eta}$.
4. *The asymptotic convergence rate of the conjugate gradient method for the preconditioned system is faster than the rate of convergence of* (4.2).

Proof. To prove positive definiteness of A consider $\mathsf{x} \in \mathbb{R}^n$. Since B is bijective, there exists $\mathsf{y} \in \mathbb{R}^n$ such that $\mathsf{y} = \mathsf{B}^{-1}\mathsf{x}$. Then

$$(\mathsf{Ax},\mathsf{x}) = (\mathsf{ABy},\mathsf{By}) = ((\mathsf{AB}-\mathrm{I})\mathsf{y},\mathsf{By}) + (\mathsf{y},\mathsf{By})$$

It is enough to show that

$$((\mathrm{I}-\mathsf{AB})\mathsf{y},\mathsf{By}) - (\mathsf{y},\mathsf{By}) < 0$$

or equivalently

$$\begin{aligned}((\mathrm{I}-\mathsf{AB})\mathsf{y},\mathsf{y})_{\mathsf{B}} - \|\mathsf{y}\|_{\mathsf{B}}^2 &< 0\\ ((\mathrm{I}-\mathsf{AB})\mathsf{y},\mathsf{y})_B &\leq \|(\mathrm{I}-\mathsf{AB})\mathsf{y}\|_{\mathsf{B}}\|\mathsf{y}\|_{\mathsf{B}}\\ &\leq \underbrace{\|(\mathrm{I}-\mathsf{AB})\|_{\mathsf{B}}}_{<1 \text{ by } (\mathrm{C.1})}\|\mathsf{y}\|_{\mathsf{B}}\|\mathsf{y}\|_{\mathsf{B}}\\ &< \|\mathsf{y}\|_{\mathsf{B}}^2\end{aligned}$$

and therefore the results follows. The fact that the iteration is convergent is obvious. In order to prove the estimate for the condition number we use the Neumann series for operator $(\mathsf{AB})^{-1}$. We have:

$$(\mathsf{AB})^{-1} = \sum_{i=1}^{\infty} (\mathrm{I} - \mathsf{AB})^i$$

Since $\|(\mathrm{I} - \mathsf{AB})\|_{\mathsf{B}} < 1$, then

$$\|(\mathsf{AB})^{-1}\|_{\mathsf{B}} \leq \frac{1}{1 - \|(\mathrm{I} - \mathsf{AB})\|_{\mathsf{B}}}$$

and therefore

$$\kappa(\mathsf{AB}) = \|\mathsf{AB}\|_{\mathsf{B}} \|(\mathsf{AB})^{-1}\|_{\mathsf{B}} \leq \frac{\|\mathsf{AB}\|_{\mathsf{B}}}{1 - \|(\mathrm{I} - \mathsf{AB})\|_{\mathsf{B}}} \leq \frac{\|\mathrm{I}\|_{\mathsf{B}} + \|\mathrm{I} - \mathsf{AB}\|_{\mathsf{B}}}{1 - \|(\mathrm{I} - \mathsf{AB})\|_{\mathsf{B}}} = \frac{1+\eta}{1-\eta}$$

Finally, the rate of convergence of the CG method for the preconditioned system is

$$\delta = \frac{\sqrt{\kappa(\mathsf{AB})} - 1}{\sqrt{\kappa(\mathsf{AB})} + 1} \leq \frac{\sqrt{\frac{1+\eta}{1-\eta}} - 1}{\sqrt{\frac{1+\eta}{1-\eta}} + 1} = \frac{1 - \sqrt{1-\eta^2}}{\eta} < \eta$$

□

C.1.1 Norm equivalence

Lemma C.1. *Assume that* A *and* B*, defined on the finite dimensional vector space* V*, are both self-adjoint and positive definite operators with respect to* $(\cdot, \cdot)$ *and* c_0, c_1 *are positive constants. Then the following are equivalent* $\forall\, v \in V$*:*

$$\begin{aligned}
c_0(\mathsf{A}v, v) &\leq (\mathsf{ABA}v, v) \leq c_1(\mathsf{A}v, v), \\
c_0(\mathsf{B}v, v) &\leq (\mathsf{BAB}\mathsf{v}, v) \leq c_1(\mathsf{B}v, v), \\
c_1^{-1}(\mathsf{A}v, v) &\leq (\mathsf{B}^{-1}v, v) \ \leq c_0^{-1}(\mathsf{A}v, v), \\
c_1^{-1}(\mathsf{B}v, v) &\leq (\mathsf{A}^{-1}v, v) \ \leq c_0^{-1}(\mathsf{B}v, v).
\end{aligned} \tag{C.2a}$$

Additionally, the condition number $\kappa(\mathsf{AB}) \leq \dfrac{c_1}{c_0}$.

Proof. The proof of the equivalence is a direct consequence of self-adjointness of A and B. For the condition number estimate consider (C.2a). Then

$$c_0 \leq \frac{(\mathsf{B}v, v)}{(\mathsf{A}^{-1}v, v)} \leq c_1$$

Let (λ_i, u_i) be the *generalized eigenpairs* of matrix B with respect to matrix A^{-1}. That is:

$$\mathsf{B}u_i = \lambda_i \mathsf{A}^{-1}u_i, \ i = 1, ..., N$$

Then (λ_i, u_i) are the eigenpairs of the matrix AB. Exploiting the properties of the Rayleigh quotient we have

$$\lambda_{\min} = \min_{v \neq 0} \frac{(\mathsf{B}v, v)}{(\mathsf{A}^{-1}v, v)} \text{ and } \lambda_{\max} = \max_{v \neq 0} \frac{(\mathsf{B}v, v)}{(\mathsf{A}^{-1}v, v)}$$

and therefore $\kappa(\mathsf{AB}) = \dfrac{\lambda_{\max}}{\lambda_{\min}} \leq \dfrac{c_1}{c_0}$. □

C.1.2 Nepomnyaschikh fictitious space lemmma

Lemma C.2. *Let X, Y, be Hilbert spaces with inner products $(\cdot,\cdot)_X$ and $(\cdot,\cdot)_Y$ respectively. Define the following two sesquilinear, continuous, Hermitian and coercive forms defined on X and Y respectively.*

$$b(x, \delta x) = \langle \mathrm{B}x, \delta x \rangle_{X' \times X} \qquad \textit{with } \mathrm{B} : X \to X'$$

$$a_0(y, \delta y) = \langle \mathrm{A}_0 y, \delta y \rangle_{Y' \times Y} \qquad \textit{with } \mathrm{A}_0 : Y \to Y'$$

Additionally we assume the existence of a continuous surjective operator $\mathrm{R} : Y \to X$ and a continuous injective operator $\mathrm{T} : X \to Y$ that satisfy

$$\mathrm{R} \circ \mathrm{T} = id_X \quad i.e. \quad \mathrm{RT}x = x \ \forall x \in X.$$

Take now an arbitrary $l \in X'$, and consider the following two variational problems:

$$\text{(C.3)} \qquad \begin{cases} \textit{Find } x \in X : \\ b(x, \delta x) = \langle l, \delta x \rangle \quad \forall\, \delta x \in X \end{cases} \quad \textit{and} \quad \begin{cases} \textit{Find } y \in Y : \\ a_0(y, \delta y) = \langle \mathrm{R}' l, \delta y \rangle = \langle l, \mathrm{R}\delta y \rangle \quad \forall\, \delta y \in Y \end{cases}$$

and assume the following two inequalities:

$$\text{(C.4a)} \qquad \exists\, c_{\mathrm{R}} > 0 : \forall y \in Y \qquad b(\mathrm{R}y, \mathrm{R}y) \leq c_{\mathrm{R}}^2 a_0(y, y) \qquad \textit{and}$$

(C.4b) $$\exists\, d_{\mathrm{R}} > 0 : \forall x \in X \quad a_0(\mathrm{T}x, \mathrm{T}x) \leq d_{\mathrm{R}}^{-2} b(x,x)$$

Now let $l \in Y'$ and x, y be the solutions of the variational problems (C.3). *Then*

$$c_{\mathrm{R}}^{-2} a_0(y,y) \leq b(x,x) \leq d_{\mathrm{R}}^{-2} a_0(y,y)$$

Proof. For the upper bound we have:

$$\begin{aligned}
b(x,x) &= \langle l, x \rangle && (1^{\text{st}} \text{ variational problem}) \\
&= \langle l, \mathrm{RT}x \rangle && (\mathrm{RT} = \mathrm{id}_X) \\
&= \langle \mathrm{R}' l, Tx \rangle \\
&= a_0(y, \mathrm{T}x) && (2^{\text{nd}} \text{ variational problem}) \\
&\leq a_0(y,y)^{1/2} a_0(\mathrm{T}x, \mathrm{T}x)^{1/2} && (\text{Cauchy-Schwarz inequality}) \\
&\leq a_0(y,y)^{1/2} d_{\mathrm{R}}^{-1} b(x,x)^{1/2} && (\text{by (C.4b)})
\end{aligned}$$

Dividing by $b(x,x)^{1/2}$ and squaring both sides gives

$$b(x,x) \leq d_{\mathrm{R}}^{-2} a_0(y,y).$$

Similarly for the lower bound we have:

$$\begin{aligned}
a_0(y,y) &= \langle l, \mathrm{R}y \rangle && (2^{\text{st}} \text{ variational problem}) \\
&= b(x, \mathrm{R}y) && (1^{\text{st}} \text{ variational problem}) \\
&\leq b(x,x)^{1/2} b(\mathrm{R}y, \mathrm{R}y)^{1/2} && (\text{Cauchy-Schwarz inequality}) \\
&\leq b(x,x)^{1/2} c_{\mathrm{R}} a_0(y,y)^{1/2} && (\text{by (C.4a)})
\end{aligned}$$

Dividing by $c_R a_0(y,y)^{1/2}$ and squaring both sides gives

$$c_{\mathrm{R}}^{-2} a_0(y,y) \leq b(x,x).$$

□

C.2 Schur complement - norm equivalence

Let U be a Hilbert space given as a Cartesian product of two Hilbert spaces U_1, U_2. Let $u = (u_1, u_2), v = (v_1, v_2)$. Assume, we are given a sesquilinear, continuous, Hermitian form:

$$b(u, v) = b_{11}(u_1, v_1) + b_{12}(u_2, v_1) + b_{21}(u_1, v_2) + b_{22}(u_2, v_2)\,.$$

Let $\mathrm{B}, \mathrm{B}_{11}, \mathrm{B}_{12}, \mathrm{B}_{21}, \mathrm{B}_{22}$ be the corresponding operators,

$$\begin{aligned}
&\mathrm{B} : U \to U' && \langle \mathrm{B}u, v\rangle && = b(u, v) && u \in U,\ v \in U\\
&\mathrm{B}_{11} : U_1 \to U_1' && \langle \mathrm{B}_{11}u_1, v_1\rangle && = b_{11}(u_1, v_1) && u_1 \in U_1,\ v_1 \in U_1\\
&\mathrm{B}_{12} : U_2 \to U_1' && \langle \mathrm{B}_{12}u_2, v_1\rangle && = b_{12}(u_2, v_1) && u_2 \in U_2,\ v_1 \in U_1\\
&\mathrm{B}_{21} : U_1 \to U_2' && \langle \mathrm{B}_{21}u_1, v_2\rangle && = b_{21}(u_1, v_2) && u_1 \in U_1,\ v_2 \in U_2\\
&\mathrm{B}_{22} : U_2 \to U_2' && \langle \mathrm{B}_{22}u_2, v_2\rangle && = b_{22}(u_2, v_2) && u_2 \in U_2,\ v_2 \in U_2.
\end{aligned}$$

Lemma C.3. *Assume additionally that b_{11} is positive definite. This implies that B_{11} is invertible and the following identity holds:*

$$\inf_{u_1 \in U_1} b((u_1, u_2), (u_1, u_2)) = \inf_{u_1 \in U_1} \langle \mathrm{B}(u_1, u_2), (u_1, u_2)\rangle = \langle (\mathrm{B}_{22} - \mathrm{B}_{21}\mathrm{B}_{11}^{-1}\mathrm{B}_{12})u_2, u_2\rangle\,.$$

Proof. Note that b being hermitian implies:

$$b_{12}(u_1, v_2) = \overline{b_{21}(v_2, u_1)}\,.$$

Taking derivative in u_1 leads to the necessary (and sufficient) condition:

$$2\Re\left(b_{11}(u_1, v_1) + b_{12}(u_2, v_1)\right) = 0 \quad \forall v_1 \in U_1\,,$$

which, in turn, implies[1]

$$b_{11}(u_1, v_1) + b_{12}(u_2, v_1) = 0 \quad \forall v_1 \in U_1\,.$$

Consequently,

$$u_1 = -\mathrm{B}_{11}^{-1}\mathrm{B}_{12}u_2\,,$$

[1] If the real part of a complex antilinear functional vanishes then so must the whole functional.

and the infimum (minimum) is equal to

$$\langle \mathrm{B}_{11}\mathrm{B}_{11}^{-1}\mathrm{B}_{12}u_2, \mathrm{B}_{11}^{-1}\mathrm{B}_{12}u_2\rangle - \langle \mathrm{B}_{12}u_2, \mathrm{B}_{11}^{-1}\mathrm{B}_{12}u_2\rangle - \langle \mathrm{B}_{21}\mathrm{B}_{11}^{-1}\mathrm{B}_{12}u_2, u_2\rangle + \langle \mathrm{B}_{22}u_2, u_2\rangle\,.$$

The first two terms cancel out and we obtain the desired result. □

Let $a(u,v)$ be now another sesquilinear, continuous, Hermitian, semi-positive form on U with a positive-definite part a_{11} as well. Denote the two Schur complement operators by:

$$\mathrm{S}_\mathrm{A} := \mathrm{A}_{22} - \mathrm{A}_{21}\mathrm{A}_{11}^{-1}\mathrm{A}_{12}, \qquad \mathrm{S}_\mathrm{B} := \mathrm{B}_{22} - \mathrm{B}_{21}\mathrm{B}_{11}^{-1}\mathrm{B}_{12}\,.$$

Lemma C.4. *Assume forms a and b (or operators A and B) are positive semi-definite and spectrally equivalent, i.e.,*

$$c_1\, a(u,u) \le b(u,u) \le c_2\, a(u,u) \quad \Leftrightarrow \quad c_1\,\langle \mathrm{A}u, u\rangle \le \langle \mathrm{B}u, u\rangle \le c_2\,\langle \mathrm{A}u, u\rangle\,,$$

with some positive constants c_1, c_2. Then the corresponding Schur complement operators are spectrally equivalent on U_2 with the same constants,

$$c_1\,\langle \mathrm{S}_\mathrm{A}u_2, u_2\rangle \le \langle \mathrm{S}_\mathrm{B}u_2, u_2\rangle \le c_2\,\langle \mathrm{S}_\mathrm{A}u_2, u_2\rangle\,.$$

Proof. We prove the lower bound.

$$\begin{aligned}
\langle \mathrm{S}_\mathrm{B}u_2, u_2\rangle &= \inf_{u_1}\langle \mathrm{B}(u_1,u_2), (u_1,u_2)\rangle && \text{(Lemma C.3)}\\
&= \langle \mathrm{B}(u_1,u_2), (u_1,u_2)\rangle && \text{(For some specific } u_1\text{)}\\
&\ge c_1\,\langle \mathrm{A}(u_1,u_2), (u_1,u_2)\rangle\\
&\ge c_1 \inf_{u_1}\langle \mathrm{A}(u_1,u_2), (u_1,u_2)\rangle\\
&= c_1\langle \mathrm{S}_\mathrm{A}u_2, u_2\rangle && \text{(Lemma C.3)}
\end{aligned}$$

The proof of the upper bound is fully analogous. □

Appendix D

Perfectly matched layer for the DPG method

Scattering wave problems, like the ones considered in Section 7.2, are usually posed in unbounded domains. However, computer simulations require a bounded domain to be defined and this can cause undesired reflections. A common technique to minimize these reflections, originally developed by Berenger in [9], is called *perfectly matched layer* (PML). It is an artificial absorbing layer surrounding the computational domain. Even though there are several other methods to model an unbounded domain, the PML is preferred because of its simplicity and accuracy.

For the numerical examples presented in this dissertation we follow the work of Vaziri Astaneh et al. [115]. In there, the authors describe the development of DPG ultraweak formulations with perfectly matched layers, using two different complex stretching strategies. Several experiments are presented verifying the efficacy of these strategies. We mention that previous work on PML for the DPG method was done by Bramwell [13].

D.1 The complex stretching function

Let Ω_c and Ω denote the computational domain and the domain including the surrounding PML region respectively, i.e, $\Omega_c \subset \Omega$. Then, the PML region is given by $\Omega_{\text{PML}} := \Omega_2 \backslash \Omega_1$. Consider now the complex stretching map $\mathbb{R}^3 \ni (x_1, x_2, x_3) \mapsto (\widetilde{x}_1, \widetilde{x}_2, \widetilde{x}_2) \in \mathbb{C}^3$ defined by

$$\widetilde{x}_k = \begin{cases} x_k & \text{in } \Omega_c \\ x_k + if(x_i, \omega), & \text{in } \Omega_{\text{PML}} \end{cases} \tag{D.1}$$

where $f(x_i, \omega) > 0$. Then, the propagating wave modes of the form $e^{i\omega \widetilde{x}_k}$ decay exponentially. In other words, the complex stretching function above is designed in such a way that it is the

identity map inside the computational domain but exponentially 'kills' the wave inside the PML region.

D.2 Model problem: linear acoustics

Consider now the time-harmonic form of the linear acoustics equations in the stretched coordinates:

$$\begin{cases} i\omega\widetilde{p} + \operatorname{div}\widetilde{u} = \widetilde{f} \\ i\omega\widetilde{u} + \nabla\widetilde{p} \quad = 0 \end{cases}$$

Using the transformation rules in the canonical Hilbert spaces given in [115], the above equations transformed in the spacial coordinates are:

$$\begin{cases} i\omega p + \ \det(J)^{-1}\operatorname{div} u = f \\ i\omega\ \det(J)^{-1}Ju + J^{-T}\nabla p = 0 \end{cases} \tag{D.2}$$

where

$$J = \begin{pmatrix} \frac{\partial\widetilde{x}_1}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial\widetilde{x}_2}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial\widetilde{x}_3}{\partial x_3} \end{pmatrix}$$

is the Jacobian of the stretching function and $\det(J)$ is its determinant. J^{-T} denotes the transpose inverse of J.

D.3 The ultraweak DPG formulation with PML

The ultraweak DPG formulation can now be derived from (D.2) as usual. Notice that it is convenient to multiply the first and second equations in (D.2) by $\det(J)$ and J^T respectively. Multiplying by test functions q and v and integrating oven the domain Ω we obtain

$$\begin{cases} i\omega(\det(J)p, q) + (\operatorname{div} u, q) \qquad = (f, q) \\ i\omega(\det(J)^{-1}J^TJu, v) + (\nabla p, v) = 0 \end{cases}$$

We follow now the standard derivation of the ultraweak formulation; we "break" the test functions and integrate by parts both equations in an element-wise fashion. The final formulation reads:

$$\begin{cases} p \in L^2(\Omega), u \in (L^2(\Omega))^3, (\hat{p}, \hat{u}_n) \in \hat{U} & \\ i\omega(\det(J)p, q) - (u, \nabla q) + \langle \hat{u}_n, q \rangle_{\Gamma_h} & = (f, q), \quad q \in H^1(\Omega_h) \\ i\omega(\det(J)^{-1} J^T J u, v) - (p, \operatorname{div} v) + \langle \hat{p}, v \rangle_{\Gamma_h} = 0, & \quad v \in H(\operatorname{div}, \Omega_h) \end{cases}$$

where the definitions of the energy spaces are given in Section 2.4.2. Above the space $\hat{U}$ is a subspace of $H^{1/2}(\Gamma_h) \times H^{1/2}(\Gamma_h)$ that incorporates the problem boundary conditions. On the outer boundary of the PML region a homogeneous soft ($\hat{p} = 0$) or hard ($\hat{u}_n = 0$) boundary condition is enforced. Finally, the optimal test norm (see adjoint graph norm in (3.16)) is also stretched and it is given by

$$\|(q, v)\|_V^2 = \|i\omega(\det(J)J^T J)^* v + \nabla q\|^2 + \|i\omega \bar{J} q + \operatorname{div} v\|^2 + \alpha(\|q\|^2 + \|v\|^2)$$

Here, $\bar{J}$ denotes the conjugate of J, and α is a scaling constant of order one.

For all the numerical experiments presented in Section 7.2 we used the stretching function given in (D.1) with

$$f(x_i, \omega) = \frac{50}{\omega} \left(\frac{x_i - l}{L - l} \right)^2$$

where (l, L) is the interval of the PML region in each space direction.

Bibliography

[1] Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., and Koster, J. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41 (electronic).

[2] Arnold, D., Falk, R., and Winther, R. (1997). Preconditioning in H(div) and applications. *Math. Comp.*, 66(219):957–984.

[3] Astaneh, A. V., Fuentes, F., Mora, J., and Demkowicz, L. (2018). High-order polygonal discontinuous Petrov–Galerkin (PolyDPG) methods using ultraweak formulations. *Comput. Methods Appl. Mech. Engrg.*, 332:686–711.

[4] Babuška, I. (1971). Error-bounds for finite element method. *Numer. Math.*, 16:322–333.

[5] Babuška, I. and Melenk, J. M. (1997). The Partition of Unity Method. *Internat. J. Numer. Methods Engrg.*, 40(4):727–758.

[6] Babuška, I. M. and Sauter, S. A. (1997). Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM J. Numer. Anal.*, 34(6):2392–2423.

[7] Barker, A. T., Brenner, S. C., Park, E.-H., and Sung, L.-Y. (2014). A one-level additive Schwarz preconditioner for a discontinuous Petrov–Galerkin method. In *Domain Decomposition Methods in Science and Engineering XXI*, pages 417–425. Springer.

[8] Barker, A. T., Dobrev, V., Gopalakrishnan, J., and Kolev, T. (2018). A Scalable Preconditioner for a Primal Discontinuous Petrov–Galerkin Method. *SIAM J. Sci. Comput.*, 40(2):A1187–A1203.

[9] Berenger, J.-P. (1994). A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185 – 200.

[10] Bernkopf, M. and Melenk, J. M. (2018). Analysis of the *hp*-version of a first order system least squares method for the Helmholtz equation. *arXiv preprint arXiv:1808.07825*.

[11] Bochev, P. and Gunzburger, M. (2009). *Least-Squares Finite Element Methods*. Vol. 166 of Applied Mathematical Sciences. Springer Verlag.

[12] Bonazzoli, M., Dolean, V., Graham, I. G., Spence, E. A., and Tournier, P.-H. (2017). Domain decomposition preconditioning for the high-frequency time-harmonic Maxwell equations with absorption. *arXiv preprint arXiv:1711.03789*.

[13] Bramwell, J. A. (2013). *A discontinuous Petrov-Galerkin method for seismic tomography problems*. PhD thesis, The University of Texas at Austin.

[14] Broersen, D., Dahmen, W., and Stevenson, R. (2018). On the stability of DPG formulations of transport equations. *Math. Comp.*, 87(311):1051–1082.

[15] Buttari, A. (2013). Fine-Grained Multithreading for the Multifrontal QR Factorization of Sparse Matrices. *SIAM J. Sci. Comput.*, 35(4).

[16] Cai, X.-C. and Widlund, O. B. (1992). Domain decomposition algorithms for indefinite elliptic problems. *SIAM J. Sci. Statist. Comput.*, 13(1):243–258.

[17] Cai, Z., Manteuffel, T. A., and McCormick, S. F. (1997). First-Order System Least Squares for Second-Order Partial Differential Equations: Part II. *SIAM J. Numer. Anal.*, 34(2):425–454.

[18] Calandra, H., Gratton, S., Pinel, X., and Vasseur, X. (2013). An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media. *Numer. Linear Algebra Appl.*, 20(4):663–688.

[19] Carstensen, C., Demkowicz, L., and Gopalakrishnan, J. (2016). Breaking spaces and forms for the DPG method and applications including Maxwell equations. *Comput. Math. Appl.*, 72(3):494–522.

[20] Carstensen, C. and Hellwig, F. (2016). Low-Order Discontinuous Petrov–Galerkin Finite Element Methods for Linear Elasticity. *SIAM J. Numer. Anal.*, 54(6):3388–3410.

[21] Cessenat, O. and Despres, B. (1998). Application of an Ultra Weak Variational Formulation of Elliptic PDEs to the Two–Dimensional Helmholtz Problem. *JNA*, 35(1):255–299.

[22] Chan, J., Demkowicz, L., and Moser, R. (2014a). A DPG method for steady viscous compressible flow. *Comput. & Fluids*, 98:69–90.

[23] Chan, J., Heuer, N., Bui-Thanh, T., and Demkowicz, L. (2014b). A robust DPG method for convection-dominated diffusion problems II: Adjoint boundary conditions and mesh-dependent test norms. *Comput. Math. Appl.*, 67(4):771–795.

[24] Congreve, S., Gedicke, J., and Perugia, I. (2018). Robust adaptive hp discontinuous Galerkin finite element methods for the Helmholtz equation. *arXiv preprint arXiv:1808.03567*.

[25] Demkowicz, L. (2007). *Computing with hp-adaptive finite elements. Vol. 1.* Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series. Chapman & Hall/CRC, Boca Raton, FL. One and two dimensional elliptic and Maxwell problems, With 1 CD-ROM (UNIX).

[26] Demkowicz, L. (2015). Various Variational Formulations and Closed Range Theorem. *ICES Report*, 15-03.

[27] Demkowicz, L. and Gopalakrishnan, J. (2010). A class of discontinuous Petrov–Galerkin methods. Part I: The transport equation. *Comput. Methods Appl. Mech. Engrg.*, 199(23-24):1558–1572.

[28] Demkowicz, L. and Gopalakrishnan, J. (2011). A class of discontinuous Petrov-Galerkin methods. II. Optimal test functions. *Numer. Methods Partial Differential Equations*, 27(1):70–105.

[29] Demkowicz, L. and Gopalakrishnan, J. (2013a). A primal DPG method without a first-order reformulation. *Comput. Math. Appl.*, 66(6):1058 – 1064.

[30] Demkowicz, L. and Gopalakrishnan, J. (2013b). An Overview of the DPG Method. *ICES Report*, 13-02.

[31] Demkowicz, L. and Gopalakrishnan, J. (2015). Discontinuous Petrov-Galerkin (DPG) Method. *ICES Report*, 15-20.

[32] Demkowicz, L., Gopalakrishnan, J., and Keith, B. (2018). The DPG-star method. *ArXiv e-prints, arXiv:1809.03153 [math.NA]*.

[33] Demkowicz, L., Gopalakrishnan, J., Muga, I., and Zitelli, J. (2012a). Wavenumber explicit analysis of a DPG method for the multidimensional Helmholtz equation. *Comput. Methods Appl. Mech. Engrg.*, 213/216:126–138.

[34] Demkowicz, L., Gopalakrishnan, J., Nagaraj, S., and Sepulveda, P. (2017). A spacetime DPG method for the Schrodinger equation. *SIAM J. Numer. Anal.*, 55(4):1740–1759.

[35] Demkowicz, L., Gopalakrishnan, J., and Niemi, A. H. (2012b). A class of discontinuous Petrov–Galerkin methods. Part III: Adaptivity. *Appl. Numer. Math.*, 62(4):396–427.

[36] Demkowicz, L. and Heuer, N. (2013). Robust DPG method for convection-dominated diffusion problems. *SIAM J. Numer. Anal.*, 51(5):2514–2537.

[37] Demkowicz, L., Kurtz, J., Pardo, D., Paszyński, M., Rachowicz, W., and Zdunek, A. (2008). *Computing with hp-adaptive finite elements. Vol. 2.* Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series. Chapman & Hall/CRC, Boca Raton, FL. Frontiers: three dimensional elliptic and Maxwell problems with applications.

[38] Demkowicz, L. F. (2008). Polynomial exact sequences and projection-based interpolation with application to Maxwell equations. In *Mixed finite elements, compatibility conditions, and applications*, pages 101–158. Springer.

[39] Dryja, M. and Widlund, O. B. (1994). Domain decomposition algorithms with small overlap. *SIAM J. Sci. Comput.*, 15(3):604–620.

[40] Ellis, T., Demkowicz, L., and Chan, J. (2014a). Locally conservative discontinuous Petrov–Galerkin finite elements for fluid problems. *Comput. Math. Appl.*, 68(11):1530–1549.

[41] Ellis, T., Demkowicz, L., Chan, J., and Moser, R. (2014b). Space-time DPG: Designing a method for massively parallel CFD. *ICES report*, 14–32.

[42] Ellis, T. E. (2016). *Space-time discontinuous Petrov-Galerkin finite elements for transient fluid mechanics*. PhD thesis, The University of Texas at Austin.

[43] Engquist, B. and Ying, L. (2011a). Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation. *Comm. Pure Appl. Math.*, 64(5):697–735.

[44] Engquist, B. and Ying, L. (2011b). Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. *Multiscale Model. Simul.*, 9(2):686–710.

[45] Erlangga, Y. A. (2008). Advances in iterative methods and preconditioners for the Helmholtz equation. *Arch. Comput. Methods Eng.*, 15(1):37–66.

[46] Ernesti, J. and Wieners, C. (2018). A space-time discontinuous Petrov- Galerkin method for acousticwaves. Technical Report 15, Karlsruher Institut für Technologie (KIT).

[47] Ernst, O. G. and Gander, M. J. (2012). Why it is difficult to solve Helmholtz problems with classical iterative methods. In *Numerical analysis of multiscale problems*, volume 83 of *Lect. Notes Comput. Sci. Eng.*, pages 325–363. Springer, Heidelberg.

[48] Feng, X. and Wu, H. (2009). Discontinuous Galerkin Methods for the Helmholtz Equation with Large Wave Number. *SIAM J. Numer. Anal.*, 47(4):2872–2896.

[49] Feng, X. and Wu, H. (2011). hp-Discontinuous Galerkin Methods for the Helmholtz Equation with Large Wave Number. *Math. Comp.*, 80(276):1997–2024.

[50] Fuentes, F. (2018). *Various applications of discontinuous Petrov-Galerkin (DPG) finite element methods*. PhD thesis, The University of Texas at Austin.

[51] Fuentes, F., Demkowicz, L., and Wilder, A. (2017a). Using a DPG method to validate DMA experimental calibration of viscoelastic materials. *Comput. Methods Appl. Mech. Engrg.*, 325:748–765.

[52] Fuentes, F., Keith, B., Demkowicz, L., and Le Tallec, P. (2017b). Coupled variational formulations of linear elasticity and the DPG methodology. *J. Comput. Phys.*, 348:715–731.

[53] Fuentes, F., Keith, B., Demkowicz, L., and Nagaraj, S. (2015). Orientation embedded high order shape functions for the exact sequence elements of all shapes. *Comput. Math. Appl.*, 70(4).

[54] Führer, T. and Heuer, N. (2017). Robust coupling of DPG and BEM for a singularly perturbed transmission problem. *Comput. Math. Appl.*, 74(8):1940–1954.

[55] Führer, T., Heuer, N., Karkulik, M., and Rodríguez, R. (2018a). Combining the DPG method with finite elements. *Comput. Methods Appl. Math.*, 18(4):639–652.

[56] Führer, T., Heuer, N., and Stephan, E. P. (2018b). On the DPG method for Signorini problems. *IMA J. Numer. Anal.*, 38(4):1893–1926.

[57] G. Graham, I., A. Spence, E., and Vainikko, E. (2015). Domain decomposition preconditioning for High-Frequency Helmholtz problems with absorption. *Math. Comp.*, 86.

[58] Gander, M. and Zhang, H. (2018). A Class of Iterative Solvers for the Helmholtz Equation: Factorizations, Sweeping Preconditioners, Source Transfer, Single Layer Potentials, Polarized Traces, and Optimized Schwarz Methods. *SIAM Rev.*

[59] Gander, M. J., Graham, I. G., and Spence, E. A. (2015). Applying GMRES to the Helmholtz equation with shifted Laplacian preconditioning: what is the largest shift for which wavenumber-independent convergence is guaranteed? *Numer. Math.*, 131(3):567–614.

[60] Gopalakrishnan, J. (2013). Five lectures on DPG methods. *arXiv preprint arXiv:1306.0557*.

[61] Gopalakrishnan, J. (Fall 2014). Geometric Multilevel Methods. *Diary of results proved and discussed in MTH 610, PSU*.

[62] Gopalakrishnan, J., Muga, I., and Olivares, N. (2014). Dispersive and Dissipative Errors in the DPG Method with Scaled Norms for Helmholtz Equation. *SIAM J. Sci. Comput.*, 36(1):A20–A39.

[63] Gopalakrishnan, J. and Pasciak, J. E. (2003). Overlapping Schwarz preconditioners for indefinite time harmonic Maxwell equations. *Math. Comp.*, 72(241):1–15 (electronic).

[64] Gopalakrishnan, J., Pasciak, J. E., and Demkowicz, L. F. (2004). Analysis of a multigrid algorithm for time harmonic Maxwell equations. *SIAM J. Numer. Anal.*, 42(1):90–108 (electronic).

[65] Gopalakrishnan, J. and Qiu, W. (2014). An analysis of the practical DPG method. *Math. Comp.*, 83(286):537–552.

[66] Gopalakrishnan, J. and Schöberl, J. (2015). Degree and wavenumber [in] dependence of Schwarz preconditioner for the DPG method. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, pages 257–265. Springer.

[67] Gopalakrishnan, J. and Sepulveda, P. (2017). A spacetime DPG method for acoustic waves. *arXiv preprint arXiv:1709.08268*.

[68] Gopalakrishnan, J. and Sepulveda, P. (2018). A Spacetime DPG Method for the Wave Equation in Multiple Dimensions. *Portland Institute for Computational Science Publications*.

[69] Green, R. E., McIntire, P., and Birks, A. S. (1991). *Ultrasonic testing*. Columbus, OH : American Society for Nondestructive Testing, 2nd ed edition.

[70] Hackbusch, W. (2016). *Iterative solution of large sparse systems of equations*, volume 95 of *Applied mathematical sciences*. Springer, Cham, second edition.

[71] Heuer, N. and Karkulik, M. (2015). DPG method with optimal test functions for a transmission problem. *Comput. Math. Appl.*, 70(5):1070–1081.

[72] Heuer, N. and Karkulik, M. (2017a). A robust DPG method for singularly perturbed reaction-diffusion problems. *SIAM J. Numer. Anal.*, 55(3):1218–1242.

[73] Heuer, N. and Karkulik, M. (2017b). Discontinuous Petrov–Galerkin boundary elements. *Numer. Math.*, 135(4):1011–1043.

[74] Hiptmair, R. and Hoppe, R. H. (1999). Multilevel methods for mixed finite elements in three dimensions. *Numer. Math.*, 82(2):253–279.

[75] Hiptmair, R., Moiola, A., and Perugia, I. (2011). Plane Wave Discontinuous Galerkin Methods for the 2D Helmholtz Equation: Analysis of the p-version. *SIAM J. Numer. Anal.*, 49(1/2):264–284.

[76] Huttunen, T., Monk, P., and Kaipio, J. P. (2002). Computational Aspects of the Ultra-Weak Variational Formulation. *J. Comput. Phys.*, 182(1):27 – 46.

[77] Ihlenburg, F. and Babuška, I. (1995). Finite element solution of the Helmholtz equation with high wave number Part I: The h-version of the FEM. *Comput. Math. Appl.*, 30(9):9 – 37.

[78] Ihlenburg, F. and Babuška, I. (1997). Finite Element Solution of the Helmholtz Equation with High Wave Number Part II: The h-p Version of the FEM. *SIAM J. Numer. Anal.*, 34(1):315–358.

[79] Keith, B. (2018). *New ideas in adjoint methods for PDEs: A saddle-point paradigm for finite element analysis and its role in the DPG methodology*. PhD thesis, The University of Texas at Austin.

[80] Keith, B., Demkowicz, L., and Gopalakrishnan, J. (2017a). DPG* method. *ICES Report*, 17-25.

[81] Keith, B., Fuentes, F., and Demkowicz, L. (2016). The DPG methodology applied to different variational formulations of linear elasticity. *Comput. Methods Appl. Mech. Engrg.*, 309:579–609.

[82] Keith, B., Knechtges, P., Roberts, N. V., Elgeti, S., Behr, M., and Demkowicz, L. (2017b). An ultraweak DPG method for viscoelastic fluids. *J. Non-Newton. Fluid Mech.*, 247:107–122.

[83] Keith, B., Petrides, S., Fuentes, F., and Demkowicz, L. (2017c). Discrete least-squares finite element methods. *Comput. Methods Appl. Mech. Engrg.*, 327:226–255.

[84] Keith, B., Vaziri Astaneh, A., and Demkowicz, L. (2017d). Goal-oriented adaptive mesh refinement for non-symmetric functional settings. *ICES Report*, 17-31.

[85] Kim, S. and Zhang, H. (2015). Optimized Schwarz method with complete radiation transmission conditions for the Helmholtz equation in waveguides. *SIAM J. Numer. Anal.*, 53(3):1537–1558.

[86] Lee, B., Manteuffel, T. A., McCormick, S. F., and Ruge, J. (2000). First-Order System Least-Squares for the Helmholtz Equation. *SIAM J. Sci. Comput.*, 21(5):1927–23.

[87] Li, X. and Xu, X. (2017). Domain decomposition preconditioners for the discontinuous Petrov-Galerkin method. *ESAIM Math. Model. Numer. Anal.*, 51(3):1021–1044.

[88] Liu, J. W. H. (1992). The multifrontal method for sparse matrix solution: Theory and practice. *SIAM Rev.*, 34(1):82–109.

[89] Mandel, J. (1994). Hybrid Domain Decomposition with Unstructured Subdomains. In *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157 of *Contemporary Mathematics*, pages 103–112. AMS.

[90] Melenk, J. (1995). *On Generalized Finite Element Methods*. PhD thesis, University of Maryland at College Park.

[91] Melenk, J. and Babuška, I. (1996). The partition of unity finite element method: Basic theory and applications. *Comput. Methods Appl. Mech. Engrg.*, 139(1):289 – 314.

[92] Melenk, J. M., Parsania, A., and Sauter, S. (2013). General dg-methods for highly indefinite helmholtz problems. *SIAM J. Sci. Comput.*, 57(3):536–581.

[93] Melenk, J. M. and Sauter, S. (2010). Convergence Analysis for the Finite Element Discretizations of the Helmholtz Equations with Dirichlet-to-Nuemann Boundary Conditions. *Math. Comp.*, 79(272):1871–1914.

[94] Monk, P. and Wang, D.-Q. (1999). A least-squares method for the Helmholtz equation. *Comput. Methods Appl. Mech. Engrg.*, 175(1):121 – 136.

[95] Mora, J. and Demkowicz, L. (2017). Fast integration of DPG matrices based on tensorization. *ICES report*, 17-26.

[96] Nagaraj, S. (2018). *DPG methods for nonlinear fiber optics*. PhD thesis, The University of Texas at Austin.

[97] Nagaraj, S., Grosek, J., Petrides, S., Demkowicz, L., and Mora, J. (2019). A 3D DPG Maxwell approach to nonlinear Raman gain in fiber laser amplifiers. *J. Comput. Phys. X*, 2:100002.

[98] Nagaraj, S., Petrides, S., and Demkowicz, L. F. (2017). Construction of DPG Fortin operators for second order problems. *Comput. Math. Appl.*, 74(8):1964–1980.

[99] Nepomnyaschikh, S. (2007). Domain Decomposition Methods. *Radon Ser. Comput. Appl. Math.*, pages 89–160.

[100] Niemi, A. H., Bramwell, J. A., and Demkowicz, L. F. (2011). Discontinuous Petrov–Galerkin method with optimal test functions for thin-body problems in solid mechanics. *Comput. Methods Appl. Mech. Engrg.*, 200(9-12):1291–1300.

[101] Oden, J. and Demkowicz, L. (2018). *Applied Functional Analysis, Third Edition*. Textbooks in Mathematics. CRC Press, Taylor & Francis Group.

[102] Paige, C. C. and Saunders, M. A. (1982). LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Trans. Math. Softw.*, 8(1):43–71.

[103] Pardo, D. (2004). *Integration of hp-adaptivity with a two grid solver: applications to electromagnetics*. PhD thesis, The University of Texas at Austin.

[104] Pardo, D. and Demkowicz, L. (2006). Integration of hp-adaptivity and a two-grid solver for elliptic problems . *Comput. Methods Appl. Mech. Engrg.*, 195(7–8):674 – 710.

[105] Paschotta, R. (2008). Article on 'Gaussian beams'. In *Encyclopedia of Laser Physics and Technology*. Wiley-VCH, ISBN 978-3-527-40828-3.

[106] Petrides, S. and Demkowicz, L. F. (2017). An adaptive DPG method for high frequency time-harmonic wave propagation problems. *Comput. Math. Appl.*, 74(8):1999–2017.

[107] Roberts, N. V., Bui-Thanh, T., and Demkowicz, L. (2014). The DPG method for the Stokes problem. *Comput. Math. Appl.*, 67(4):966–995.

[108] Roberts, N. V. and Chan, J. (2017). A geometric multigrid preconditioning strategy for DPG system matrices. *Comput. Math. Appl.*, 74(8):2018 – 2043.

[109] Roberts, N. V., Demkowicz, L., and Moser, R. (2015). A discontinuous Petrov–Galerkin methodology for adaptive solutions to the incompressible Navier–Stokes equations. *J. Comput. Phys.*, 301:456–483.

[110] Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition.

[111] Sheikh, A. H., Lahaye, D., and Vuik, C. (2013). On the convergence of shifted Laplace preconditioner combined with multilevel deflation. *Numer. Linear Algebra Appl.*, 20(4):645–662.

[112] Stolk, C. C. (2013). A rapidly converging domain decomposition method for the Helmholtz equation. *J. Comput. Phys.*, 241:240–252.

[113] Stolk, C. C., Ahmed, M., and Bhowmik, S. K. (2014). A multigrid method for the Helmholtz equation with optimized coarse grid corrections. *SIAM J. Sci. Comput.*, 36(6):A2819–A2841.

[114] Tezaur, R. and Farhat, C. (2006). Three-dimensional discontinuous Galerkin elements with plane waves and Lagrange multipliers for the solution of mid-frequency Helmholtz problems. *Internat. J. Numer. Methods Engrg.*, 66(5):796–815.

[115] Vaziri Astaneh, A., Keith, B., and Demkowicz, L. (2018). On perfectly matched layers for discontinuous Petrov-Galerkin methods. *Comput. Mech.*

[116] Vion, A. and Geuzaine, C. (2014). Double sweep preconditioner for optimized Schwarz methods applied to the Helmholtz problem. *J. Comput. Phys.*, 266:171–190.

[117] Xu, J. (1992). Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613.

[118] Xu, J. (1997). An introduction to multilevel methods. In *Wavelets, multilevel methods and elliptic PDEs (Leicester, 1996)*, Numer. Math. Sci. Comput., pages 213–302. Oxford Univ. Press, New York.

[119] Xu, J. (2001). The method of subspace corrections. *J. Comput. Appl. Math.*, 128(1-2):335–362. Numerical analysis 2000, Vol. VII, Partial differential equations.

[120] Yosida, K. (1980). *Functional Analysis*, volume 123 of *Grundlehren der Mathematischen Wissenschaften (Fundamental Principles of Mathematical Sciences)*. Springer-Verlag, 6nd ed edition.

[121] Zepeda-Núñez, L. and Demanet, L. (2016). The method of polarized traces for the 2D Helmholtz equation. *J. Comput. Phys.*, 308:347–388.

[122] Zitelli, J., Muga, I., Demkowicz, L., Gopalakrishnan, J., Pardo, D., and Calo, V. M. (2011). A class of discontinuous Petrov-Galerkin methods. Part IV: the optimal test norm and time-harmonic wave propagation in 1D. *J. Comput. Phys.*, 230(7):2406–2432.